ADDIS ABABA UNIVERSITY
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Machine Learning Interfaces for Optimal Design and Control of Solar Thermal Systems in Process Industry

By

Fitsum Bekele

Supervisors

Ramchandra Bhandari (PhD), Professor, TH Köln (University of Applied Science), Germany.

Mengesha Mamo (PhD), Associate Professor, Addis Ababa University, Institute of Technology.

Dissertation submitted to School of Electrical and Computer Engineering, Addis Ababa Institute of Technology Graduate Studies, Addis Ababa University, in partial fulfilment of the requirements for the Degree of Doctor of Philosophy (PhD) in Electrical Engineering (Control Engineering).

June, 2022

Addis Ababa

ADDIS ABABAUNIVERSITY
SCHOOL OF GRADUATE STUDIES

MACHINE LEARNING INTERFACES FOR DESIGN AND CONTROL OPTIMIZATION OF SOLAR-ASSISTED INDUSTRIAL ENERGY SYSTEM

BY
FITSUM BEKELE

Declaration

This dissertation is my original work, was not copied, has not been presented for a degree in any other university, and all the source have been duly acknowledge.

Fitsum Bekele		01.06.2022
PhD candidate's Name	Signature	Date

This dissertation has been submitted with my approval as a supervisor

Ramchandra Bhandari (PhD)		01.06.2022
Supervisor's Name	Signature	Date:
Mengesha Mamo (PhD)		01.06.2022
Supervisor's Name	Signature	Date:

ADDIS ABABAUNIVERSITY

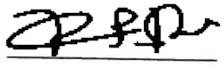
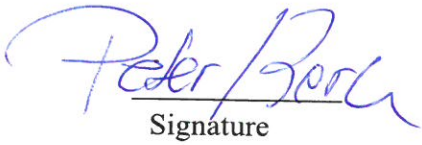
SCHOOL OF GRADUATE STUDIES

MACHINE LEARNING INTERFACES FOR DESIGN AND CONTROL OPTIMIZATION OF SOLAR-ASSISTED INDUSTRIAL ENERGY SYSTEM

BY

FITSUM BEKELE

SIGNED BY THE EXAMINING COMMITTEE:

Dereje Shiferaw (PhD) Examiner's Name	_____ Signature	01.06.2022 Date:
Getachew Bekele (PhD) Examiner's Name	 Signature	01.06.2022 Date:
Peter Kern (PhD) Examiner's Name	 Signature	01.06.2022 Date:

Abstract

The trend of introducing solar thermal systems (STSs) in process industries has resulted in a new energy paradigm– an interactive platform where there are economic benefits and motivations to address sustainable development. On the other hand, this paradigm has also introduced fluctuations and uncertainties not previously seen on the energy system, and is challenging the industry. Accordingly, we are observing increasing need for robust design and control solutions that will facilitate the smooth operation and cost competitiveness of the industrial solar thermal system (ISTS). The possibility for developing such a solution exists, but only if the necessity to explicitly model, which might not work well, and also increase computational complexity in the ISTS, is removed. In this dissertation work, a machine learning (ML) approach is followed for design and control optimization of ISTSs, and is leveraged for two goals. First it is used as a multi-modelling tool for developing heterogeneous optimization interfaces, using stochastic and generic models. These interfaces are intended to be simple but are not simpler in order to simultaneously address both scalability-tractability tradeoffs and model inefficiencies of conventional methods. Afterwards, ML enabled linking up of these interfaces as building blocks for realizing a modular optimization framework, and of integrating different layers of functionalities. As a result, the ML approach allowed disaggregated modelling of several similar technologies (and processes) as well as parameterizing of their inputs and local condition differently. Using this method, it was also possible to represent distributed energy resources (DERs) and their additional capabilities of interactions. Furthermore, it allowed replication of simulation experiments with the same model and at varying scale levels. These are essential features that cannot be offered by conventional methods, and can be used to improve synergy and unlock the potentials of DERs in ISTSs.

Following the ML approach, some important findings were made. Firstly, the solutions of the optimal design problems were scalable and tractable. This feature facilitates operation-based designs of STSs according to the specific requirements of process(s), heat distribution networks or existing thermal plant in industry. The approach also allowed the testing of an improved optimal control strategy, while at the same time, enabling controller tuning or model calibration. This capability is used to adapt an empirical solar radiation model, to serve as an efficient and low-cost sensor that can be integrated to ISTSs in real-time. However, due to the scope and limitation of the dissertation, these relevant findings provide mainly key design and control strategies and points of discussion instead of benchmarked results. Therefore, it is particularly desirable if further research could confirm these findings.

Keywords: Design optimization, optimal control, model calibration, machine learning, solar thermal system, process industry.

Acknowledgements

First and Foremost my heartfelt gratitude goes to Prof. Ramchandra Bhandari for his support and guidance throughout this time. In addition to his patience and valuable knowledge complementing this dissertation, he thought me to focus on the solution in every challenge. I also offer my heartfelt thanks to Dr. Mengesha Mamo, for his valuable assistance and suggestion, which improve not only the research approach but also clarify its objective and contribution.

I would like to especially thank Professor Benjamin L. Lamptey, who helped me get high resolution metrology data for the development of the solar radiation prediction model. With his personal interest in machine learning, his support went beyond helping me get data to actually discussing my work and giving valuable suggestions. I would also like to express my gratitude towards members of the TH Köln research platform at STEPS, for offering valuable comments on my work as well as helping me publish. I also thank colleagues and professors at AAIT for their motivating discussions.

Finally, I thank my wife, Nardos Admasu, for supporting me during my studies and this work. I would also like to include my kids, Leo & Mabelle, for the joy and emotional support.

Fitsum Bekele, June, 2022.

This dissertation was possible through the support of the Ethiopian Ministry of Education (MoE) and the German Academic Exchange Service (DAAD) under the framework of the joint Ethio-German PhD program.

Contents

List of Symbols and Abbreviations

Abbreviation	Description	1st Definition
AI	Artificial Intelligence	page 18
CC	Control Concept	page 46
DDPG	Deep Deterministic Policy Gradient	page 43
DLSTM	Deep Long Short-term Memory	page 53
DNN	Deep Neural Network	page 69
SQN	Deep Q-learning Network	page 54
DSM	Demand Side Management	page 54
DPG	Deterministic Policy Gradient	page 38
DERs	Distributed Energy Resources	page 54
DSF	Distributed Solar Field	page 8
DP	Dynamic Programming	page 59
FLC	Fuzzy Logic Controller	page 30
HRSG	Heat Recovery Steam Generator	page 62
HTF	Heat Transfer Fluid	page 47
LCOE	Levelized Cost of Energy	page 46
ML	Machine Learning	page 6
MDP	Markov Decision Process	page 28
MABE	Mean Absolute Bias Error	page 74
MPC	Model-Based Predictive Control	page 13
MC	Monte Carlo	page 30
MIMO	Multi-Input Multi-Output	page 52
MOO	Multi-Objective Optimization	page 11
NNs	Neural Networks	page 18
NARX	Nonlinear Autoregressive with Exogenous Input	page 73
PPO	Proximal Policy Optimization	page 55
PTC	Parabolic Trough Collector	Page 63
RBFN	Radial Basis Function Network	page 23
RL	Reinforcement Learning	page 19
RMSE	Root Mean Squared Error	page 74
STS	Solar Thermal System	page 6
TD	Temporal Difference	page 30
TES	Thermal Energy Storage	page 9
TRPO	Trust Region Policy Optimization	page 55

List of Figures

List of Tables

Chapter 1

Introduction

Distributed solar energy generation and its integration to industries has been shown to offer great potential in terms of creating an energy efficient industrial sector. Some of this are mitigating price volatility of imported oils as well as unreliability of the electricity grid and energy loss, and ultimately to the larger sustainable development as a whole [1]. solar thermal systems (STSs) may offer a reliable, clean, and resilient energy supply platform, leading to a decarbonized industry.

However, the trend of STSs integration introduces fluctuations and uncertainties that impact the economic feasibility and reliability of energy delivery to the process industry. This variability, is due to the nonlinear interactions of distributed resources and the stochastic nature of energy generation as well as consumptions. Recent researches suggest that the possibility for design and control concept optimizations [2], real-time forecasting of demand and generation [3] as well as optimal control [4], could enhance reliability and efficient utilization of energy in industrial solar thermal systems (ISTSs). Unfortunately recreating the behavior of these types of energy systems, for robust optimization formulations, is not possible through monolithic modeling of its constituent parts (as is done by conventional methods).

On the other hand, a machine learning (ML) approach, as is done elsewhere [5], could enable parameterizing similar STS technologies differently while including economic, and other performance measures (e.g. emission reduction) in the design problem. This way, ML allows evaluation of the ISTS from a more systemic perspective while being able to integrate individual models. Considering the energy system as a whole help derive insights for optimization of its design and operation such as through assessment of different scenarios of configurations and operations. In addition, ML-based designs adaptively evolve as influences change, thereby facilitating effective decision making processes.

The other main advantage to which ML can be applied is to enable improved synergy between the STS and the industrial process (or heat distribution network and industrial thermal plant). For example, to ensure the optimum level of operation of STSs, specific set-points are required to be established to sustain the annual usable heat at a maximum while fulfilling all operational constraints. Such adaptive operation is based on flexible utilization of all distributed energy resources (DERSs). Therefore, modelling of dynamic behaviors as well as dependencies between successive time steps and forecasting of uncertainties should be considered [6].

In other energy systems, ML is successfully implemented to handle various tasks such as optimal voltage control in active distribution network [7], demand response in smart grid [8], load frequency control [9], and building energy optimization [10]. Similar performance improvement level as of these implementations could also be achieved for ISTSs. Thus, the primary aim of this research work is to use a ML approach to solve optimization problems in ISTSs, while also developing heterogeneous model interfaces. To this end, this chapter starts by providing the state of the art in the investigation and integration efforts of STS to the process industry. This chapter also highlight the work in this PhD research, its objective, approach and intended contribution to the research arena.

1.1. Challenges of the Solar Thermal to Industry Integration

The thermal energy system in process industry, without STS, is composed of different parts that allow for the production, distribution and consumption of heat energy. Industrial boilers produce saturated steam using electricity or other energy sources such as diesel or furnace oil. For distribution, different kinds of thermal networks comprising main line, sub-lines and equipments are used. A common header can reduce the steam pressure at a given point, as per requirement, in order to supply heat for different processes. According to its state of operation, the heat demand of a process vary, and is conveyed either indirectly (via internal exchangers) or directly (using steam injection). Due to the efficiency of transport, steam injection is mostly used to heat industrial processes.

Following the general trend of introducing clean energy sources, STSs are being integrated to process industries in the potential regions (Figure 1.1) [11]. These systems are suitable, as the industrial energy generation and utilization match the climate condition, and therefore a significant exploitation possibilities exist. However, this inclusion of fluctuating energy sources

adds an additional variablity to the the industrial supply system, which impacts directly its performance.

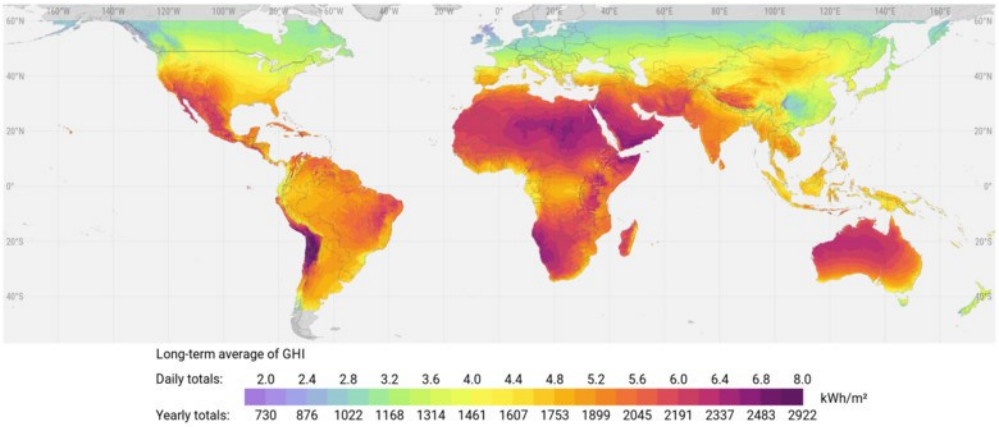


Figure 1.1. Potential regions for solar thermal integration to process industries [11].

STS in industry is an integrated energy system consisting of interconnected and distributed energy resources (DERs), which may operate in parallel with heat distribution network or in process-agumenting mode [12] (Figure1.2). STS may also be hybridized with conventional industrial plant in fuel conservation or power boosting mode, to heat part of the feed water (100 -250 ºC) or directly supply steam (300-500 ºC) (Figure1.3) [13]. This shows that STSs are not only suitable for heating individual or connected processes in supply networks, but also for integration into existing cogeneration plants. Therefore STSs could offer a range of viable industrial energy solutions in the near future.

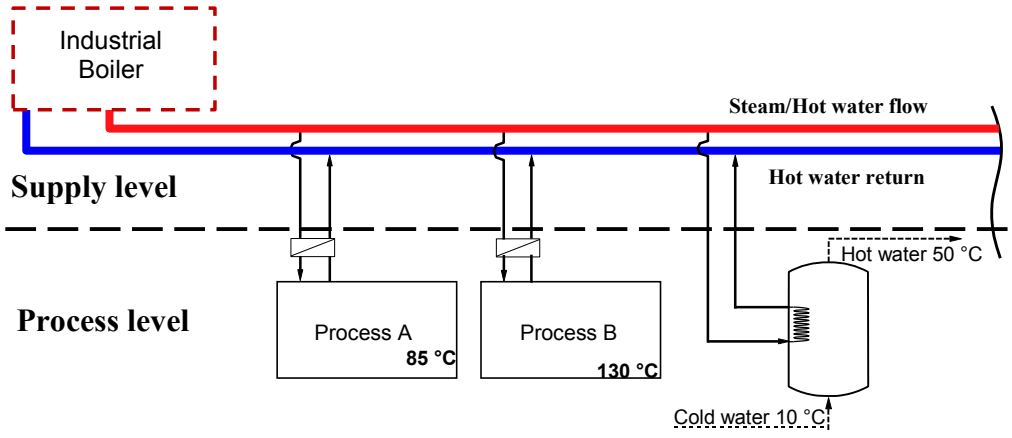


Figure 1.2. Industrial boiler and processes coupled to heat distribution network

These STS-based energy solutions, have been traditionally designed with safe margines (e.g. oversized solar field and/or storage) or with readily acting generation units (e.g. auxilary gas boiler). However, the more economic the system design, the more challenging it becomes to optimally control, as the compensating capabilities of large systems are not available. Thus, a sub-optimal operation of one DER compnent implies significant capacity drop relative to the total solar gain utilized by the system. In particular, connecting several loads, such as batch processes may cause even more capacity reduction over time.

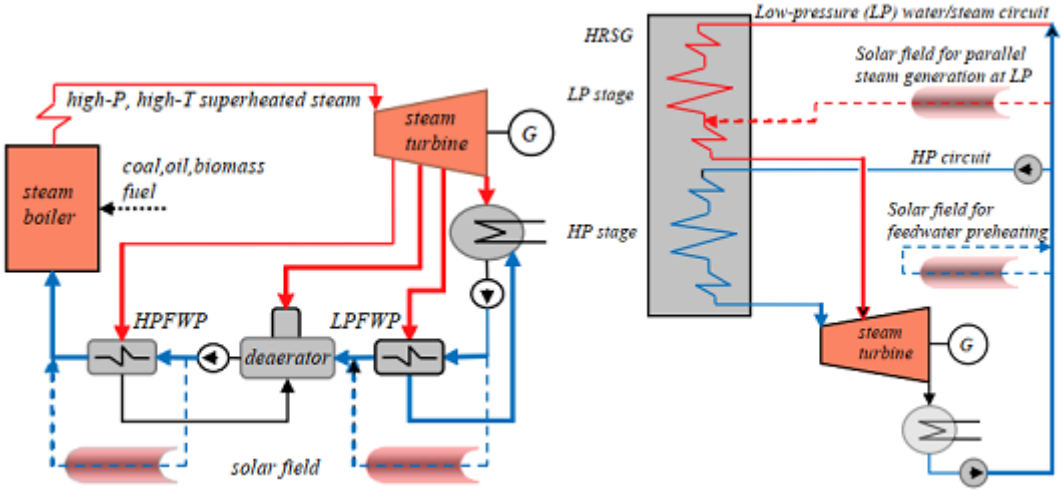


Figure 1.3. Solar steam integration schematics for (a) feed water heating (LP and HP), and (b) feed water heating and steam generation

Therefore, without STSs, thermal energy is generated and supplied at high or medium pressure/temperature levels and consumed from the distribution system. Due to the introduction of distributed STSs, thermal energy can be integrated at almost any point of the heat supply system. Also, and in order to better match the fluctuating production, auxilary heat sources, thermal energy storage (TES) and load magment schemes are being incorporated. Therefore, the operation of ISTS will involve a higher degree of complexity than the operation of the conventional supply system. To handle this challenge, remote monitoring and control of DERs (solar fields, TESs) and processess are used by implemented STS projects in industries. Therefore, STS-to-industry integration require inclustion of additional devices and capabilities such as smart sensors, actuators and controllers as well as communication networks [14]. Thus, there is a change from hierarchial operations towards an intelligent and distributed heat network in ISTSs. As a result, the challenges of integrating STSs to proces industries mean that they are

implicitly considered to be intelligent networks. Machine learning can support this change, by being an adaptable interface of this industrial heat network.

1.2. Design and Control Optimization: A review

This section discuss previous studies regarding the design and optimization of ISTSs. As we will see however, there are only few examples, in which a holistic optimization approach has been applied to the design and control of these systems. Therefore, some of the main aspects of optimization relevant to these applications will be discussed. The review is also intended to establish the motivation in describing the ITSS through its complexity using a ML approach.

As stated, ISTSs are evolving into complex energy networks not only due to generation and demand variability but also to the number and interaction of DERs. Thus, the high penetration of STS in industry may lead to fluctuations and uncertainties, which would impact energy reliability significantly. Furthermore, there is often time difference between generation and consumption as well as off-peak and peak time energy cost variations in ITSTs. A strategy to address these variability issues is to integrate auxiliary heat sources and TESs. This way, desirable features such as smoothing out the effect of weather intermittencies as well as adjusting generation interval to peak load hours could be attained. Implementing this strategy, however, results in high investment, which reduce energy cost competitiveness of STSs [15]. Therefore, design optimization in the sense of technology selection, configuration and sizing is particularly desirable for the economic feasibility of STSs in process industry.

Prompted by this need, there have been considerable efforts to study the optimal design of ISTSs. For examples, Tobias et al. investigated optimization of STS for integration in a medium sized Scottish brewery [16]. The design problem was to find optimal collector and storage volume that minimize the annualized investment and operation cost of the STS with or without renewable heat incentives (RHIs). The study found out that the payback period and CO_2 saving potential depend on the technology type and RHI scheme. On the other hand, similar works that improve economic models through inclusion of other cost related aspects into the design optimization procedure were also proposed. Some cost related factors like energy loss in pump, heat exchanger, storage and pipe by Chaimaa and Arifeen [17] show the effect of location on expected solar gain in similar design variants. The other widely used economic model in design

optimization is the levelized cost of energy (LCOE). The LCOE is defined as a ratio of total life cycle cost to total lifetime energy production. Tian Z. et al. [18] employ LCOE in optimizing a hybrid solar plant for district heating. Their work showed that hybridization of the solar plant enabled reduction in the LCOE cost of heat by 5-9% when compared to the solar alone system.

The above discussed methods can be classified as cost-based optimization schemes, whereby objective functions, payback period, annualized cost or LCOE, were formulated to design economically feasible solutions. In this regard, although these cost-based schemes could be effective in optimal design, they completely depend on the availability of valid data and assumptions about cost of DERs' and their operations. In general, the number of implemented ISTSs is significantly low, with only few installations worldwide [19]. Thus, some cost structures considered in the literature are based on authors' own assumptions, and vary significantly from author to author. As a result the performance of such economic models are controversial since it is not possible to make a systematic comparison among them. Moreover, cost learning curve of technologies is often not considered in these studies. Therefore, relying on the research findings of a few years back could lead to a significant cost overestimation for the same amount of energy delivered from the same system. Consequently, future energy planning based on the output of these studies might be misleading.

To overcome the limits in the above mentioned methods, energy-based design optimization schemes have been suggested as a potential solution. For example, Bahlawan H. et al. [21] considered an objective function that minimizes cumulative energy use of a hybrid plant for optimizing its configurations. The study showed that design optimization resulted in primary energy saving in all configurations, but the extent of saving depend on the scheme of configuration. A multi-objective optimization (MOO) procedures that combine contradicting cost and energy criteria are also reported in the literature. In this approach, a feasible optimal solution set is generated to form a Pareto frontier diagram, for selecting a tradeoff optimal point. MOO based investigations such as by Habibollahzade A. et at. [22] and Ghasemi A. et al. [23] showed the effect of optimal design on hybrid plant efficiency, LCOE and co_2 emission reduction performance. All of these studies relied on a steady state models and a single operating point is used as a basis for the optimal design.

While energy-based designs perform better than economic models, modifications for control concept optimization is not straightforward, due to the modeling and simulation tools they used. Review of existing tools for the design optimization are conducted in [24] for process heat and for solar steam industrial applications in [25]. These works showed that modelling ISTSs is challenging due to complexity of interaction and fluctuations as well as the need for integrating various functionalities in the design optimization. Existing platforms restrict modelling by providing several similar STSs and processes the same inputs; and allowing interactions only at low resolutions time steps. However, this has many limitation in design optimization of STS. Most notable is the difficulty of analyzing performance changes and impacts of design variants (on existing industrial process and /or heat distribution network). Unfortunately, input parameters and local conditions change in similar DER technologies and processes, and therefore, they are both heterogeneous and uncertain in essence. This is evident from the monitored SSTS projects such as done by Wagner & Co. Solartechnik GmbH [6]. These projects, in addition, underlined the relevance of scenario-based simulations for developing a holistic design optimization framework. This framework allow construction of design variants, whereby key design variables are changed simultaneously while including additional measures (e.g. cost and/or emission).

The other dimension of challenge in ISTSs is ensuring optimal operation while fulfilling design constraints [26]. This is required in order to supply available energy at desired temperature, thereby reducing auxiliary demand and improving overall efficiency. This is also required because of the effects of multiple disturbances, leading to cyclic and wasteful startups and shutdowns of boiler. Additional capabilities that enhance energy cost competitiveness of the STS through capacity factor and productivity improvement possibilities also contribute to the requirement. These capabilities are intended to prolonged generation even after sundown as well as enable near design limit operation. Thus, a control strategy that adequately capture uncertainties while simultaneously addressing both operating point improvement and thermal loss reduction is desirable for the efficient operation of STS. Camacho [27] claimed that such control formulation should have the following desirable features. First, it must balance the trade-off between multiple goals, such as tracking desired outlet temperature and maximizing thermal energy output based on a cost function. Second, it must learn to adapt to simultaneous disturbances and model uncertainties; and third, it must consider state and input constraints for

safe operation of ISTSs. Unfortunately, these requirements are difficult to achieve using conventional control technique that operating without compensator in control loop [27].

Rule-based prescriptive control schemes use simple feedback control, and are realized in two stages: (1) formulate pre-determined options to select control set points (e.g., temperature set point), and (2) use techniques such as PID control to track the set points. These approaches are simple and can perform well around nominal operating points but they are not optimal, for two reasons. First, predictive information is not taken into consideration, leading to sub-optimal performance. Designed to deal with some aspects STS, these controllers are either tightly tuned or detuning with low gain, which result in sluggish or oscillatory response. Second, the fixed and predetermined control sequence does not address the requirements of different STS configurations that operate in other similar industries with different process profiles.

On the other hand, model-based predictive control (MPC) scheme, since its inception in 1970s, continue to result in breakthroughs across many fields. This is the motivation behind the development of MPC with additional feedforward compensator for optimal control of STSs in process industries. The compensator is often designed based on insights derived from adaptive, nonlinear or robust control fields. All MPC techniques for STS, despite difference in interpretation and implementation, basically consists of three connected steps. First, an explicit model, which can be reduced or nonlinear, characterizing the thermal energy dynamics of STS is developed. Then there is a predictive mechanism for disturbance prediction such as solar radiation and ambient temperature varations. Finally, the developed model together with the predictions determine future control sequence by minimizing a certain cost function.

There are several published works that investigate optimal control of STS in process industries using MPC method. For examples, in [28] nonlinear MPC based on feedback linearization was proposed for optimla control of STS that minimize energy loss. The schem use change of variables to transform nonlinearties into convex optimization problem that is solved using mixed integer quadratic programming. The authors claimed that the method result in less constraint violations and reduced waste energy when compared to a general-purpose nonlinear controller. In [29],centralized and logic-based decentralized MPC controllers was designed based on a model of the collector field ACUREX, located in Spain. The aim of the first controller is to optimize thermal power without a set-point tracking; whereas the second controller attempts to address the

computational effecency problem of the central (first) controller. In [30], a robust MPC scheme was investigated using Iterative Extended Kalman Filter (IEKF) for inlet temperature and solar radiation estimates. The study claimed that the approach resulted in reduced sensor cost and measurnmrnt noise. Takagi-Sugeno fuzzy that combined a parametric uncertainty model as a robust control scheme was suggested in [31]. The aim was to address both the nonlinearities of the process and disturbances act on the solar field. A fuzzy robust PI Kharitonov's and Lyapunov's stability term are used while tuning the controller using an objective function.

Although the aforementioned MPC variants are effective, there remains a lot more to do to improve their applicability for real-time optimal control of ISTSs. The solar field parameters in STS are distributed with dynamics that depend on space and time, resulting in both a nonlinear and time variant plant model. In general, these model-based opotimal control methods cannot take all aspect of the real physical systems whose underling processes is complex, and therefore make simplifying assumptions. Furthermore, these schemes are labor-intensive and are expensively calibrated by domain experts.

1.3. This Research Work

Use of several design and control optimization schemes have been proposed for ISTSs. However, existing methodologies fail to integrate all the relevant factors, holistically. During the course of this dissertation, some major research questions (gaps) has been identified.

- How do processes in heat distribution network behave for optimal STS integration?
- How to manage autonomous STS in process industry for adaptive mode of operation and flexible resource utilization?
- Does control concepts and DSM impact solar gain under stochastic demand and generation behaviors?
- How to carry out scenario-based design optimization while ensuring tractable and scalable optimal solution?
- Does learning to optimize ISTS's operation lead to better prediction of uncertainties for calibration of controllers and/or disturbance modeling?

These research gaps poses the question of whether there is a need for a different view of optimization anticipating that STS-to-industry integration challenges exist, and are, relevant. This

is especially true, if there is a need for improved learning and evidence to design as well as remotely monitor and control of DERs in ISTS. These DERs have heterogeneous behavior and nonlinear interactions, which limit accurate modeling of their dynamics. Therefore, the ISTS cannot be described by aggregating different individual DERs, if a scalable and computationally tractable optimal solution is required.

Therefore, a ML approach for hybridization of models and analysis techniques has been chosen in this work. The approach used ML as a multi-modelling tool for representing heterogeneous behaviors and interactions of DERs, and integrating this with design and control optimization procedures. In order to emulate the complex ISTS dynamics under distributed measures, optimization interfaces are developed using generic and stochastic models. Afterwards, these interfaces are used as building blocks to develop a modular optimization framework that integrate different layers of functionalities. The models interfacing the optimization procedures are intended to be as simple as possible in order to allow assessment of different scenarios of configurations and operations within reasonable time. This approach led to the achievement of important findings from the perspective of design and control optimization. In particular, it enable disaggregated modelling of several similar technologies (and processes), and parameterizing of their inputs and local condition differently. Furthermore, it allowed conducting of simulation experiments at varying scale levels with the same model.

Yet, this approach has also required simpler models that generalize well and identification of efficient techniques for metadata and heterogeneous modelling. It is worth mentioning that selection of the "right" ML algorithms and models is a complex process by itself, because it is not known a priori how to configure the algorithm parameters and what effect these configurations have on the task at hand. For such applications, it is not sufficient to use heuristics and best practices, derived from the specific approaches these algorithms were modelled in. In this dissertation, well-performing ML algorithms and models are selected through a data-driven decision process (DDP) using automated design of experiment. In this sense, DDP was used to identify both the various algorithms doing the same task (such as performance characterization, parameter estimation, uncertainty management and active data selection) as well as possible ways of measuring their performances. In addition, it is used to compare the accuracy and generalization capability of the proposed approach with other methods reported in the literature.

1.3.1. Objective

The general objective of the research is to represent the variability, interconnectedness and complexity of generation and demand in ISTSs as well as simplify their optimal design and operation for sustainable energy supply. To successfully address these requirements, the following specific objectives are also considered.

Specific objectives.

- Develop efficient model interfaces and formulate robust design optimization procedures.
- Design constrained optimal control for joint resource and operation management.
- Develop a computationally efficient and low-cost solar radiation prediction scheme using ML

1.3.2. Scope and Contribution

This dissertation deals primarily with the introduction of heterogeneous model interfaces into conventional optimization techniques and formulation of a general optimization framework. The optimization interfaces are designed to emulate evolution of the complex spatiotemporal dynamics in ISTS under distributed measures. As such, ML is used for uncertainty management and metadata modeling while developing generic and stochastic optimization interfaces. These interfaces are then used as building blocks to develop a general optimization framework a using modular and bottom-up approach. Moreover, considering the constraints for adequate and valid data within the research work schedule, it was reasonable to have to work with modular models that can be extended and modified easily. In relation to the optimization task, the dissertation also elaborate on a different and suitable solar radiation prediction scheme. However, this activity is limited only to the design of a virtual solar radiation sensor. Hence, its integration for energy management or predictive control is not covered.

It is essential to point out that the dissertation results are intended to support recommendations and serve as a basis for further research. With the stated approach and scope in mind, the dissertation address some of the research gaps and add its part to the available methods through its contributions. Although there are many examples of applied ML for real world applications, only few evaluations of the method can be found for energy systems, especially for STSs in process industry. Therefore, the application of the approach in this research work contributes

through establishing the bases of a generally accepted role of ML in optimization. Additionally, the dissertation has the following contributions:

- ML is leveraged for active dataset selection, high resolution synthetic data generation as well as construction of stochastic consumption models (through building a class of statistical models and jointly learning their operational parameters). These ML interfaces enable evaluation of different scenarios of STS configuration and operation within reasonable time horizons. This has removed the limits of conventional approaches in terms of computational tractability and scalability of optimal design solutions.
- An elegant and simple as well as new method of solar radiation prediction through empirical model calibration using reinforcement learning is developed.
- Finally, a prior knowledge about system dynamics and design of effective reward-shaping signal, to reduce long training time and improve stability (convergence) issues in a ML-based optimal control is elaborated.

To establish these claims, several case studies, through simulation experiments, are carried out using valid datasets obtained from real-time measurements in industry and metrology stations.

1.3.3. Outline

The dissertation is organized as follows. Chapter two elaborate on the design of efficient model interfaces for design and control optimization task. Heterogeneous and multi-scale modelling approach encompassing stochastic environmental and demand models as well as parameter estimation and uncertainty management are also presented. A holistic design optimization scheme integrating several requirements is considered in chapter three. Two case studies, one for solar process heating and another for hybrid solar-biomass cogeneration is presented including economic feasibility analysis. In chapter four, deep neural networks and reinforcement learning for a general intelligence framework is designed. The method allow real-time control and/or operation optimization. Furthermore, the framework is used to adapt an empirical solar radiation model for designing an efficient and low-cost virtual sensor. In the last chapter, some relevant findings are discussed to highlight the advantages and practical aspects of using the selected approach. Finally, recommendations for future research possibilities are given.

Chapter 2

Modelling and Optimization using Machine Learning

Heterogeneous multi-scale modelling is an emerging field of research for studying complex systems through representing these systems at different scales (e.g. structural, spatial and temporal). These kinds of systems depict essential features in complex systems that relate to the aim of the modelling and as such need to be represented on multiple scales with feedback loops. The modelling approach enable scalable and computationally tractable solution in optimization of STS in process industries. So, for example, conducting simulation experiments at varying scale levels with the same model offer several advantages. The requirement for such simulation capability in the design and control optimization problem comes from:

- Solar thermal energy can be utilized at almost any level (processes, conventional thermal plant and heat distribution networks) of the industrial supply system.
- Fluctuating and periodically available generation from STS imply that the solar gain varies as the industrial demand vary in time and magnitude. Demand profile can be manipulated with supply-demand interaction at the process level (using demand side management), leading to effects at the system level.
- Control concepts that are implemented, for adaptive mode of operation and flexible utilization of DERs, at lower scales will allow the analysis and optimization of operation at higher supply level.

The approach will serve as a useful tool, as it enable capture stochastic behaviors as well as allow inclusion of uncertainty in DER models. In particular, local effects in DERs comprising STS can be represented as uncertainty so that their effects can be analyzed at the system level. On the other hand, inclusion of stochastic models of energy generation and consumption is also necessary. Furthermore, representing the variety of interactions and feedbacks inherent in DERs is possible through this modelling approach. These capabilities are important because there is still a need to

improve the energy cost-competitiveness of STSs in industry, which monolithic models used by conventional methods fail to address.

2.1. Description: What is Machine Learning?

To explain the role and application of machine learning, its close relationship with artificial intelligence (AI), neural networks (NNs) and deep learning must be considered. Regarding ML from a multi-disciplinary perspective can help us to understand the various algorithms and models used by it. Considering this multi-disciplinary perspective, and maybe because of it, common roles and application for ML remains an open question. Therefore, ML is applied in many different domains and encompasses many different interpretations, reaching statistics to probability as well as game theory and neurobiology. Despite the trend of embedding more and more learning algorithms in technology, to solve or perform challenging tasks efficiently, a complete recognition of ML as a separate field does not yet exist [32]. The following statements in academic literature may provide an overview of the range of roles and applications of machine learning. According to different authors, ML is a:

- "One of the most interesting features of machine learning is that it lies on the boundary of several different academic disciplines, principally computer science, statistics, mathematics, and engineering. …machine learning is usually studied as part of artificial intelligence, which puts it firmly into computer science …understanding why these algorithms work requires a certain amount of statistical and mathematical sophistication that is often missing from computer science undergraduates", Marsland [32].
- "Vast amounts of data are being generated in many fields, and the statistician's job is to make sense of it all: to extract important patterns and trends, and to understand what the data says. We call this learning from data", Trevor H. et.al [33]."

Marsland's definition of machine learning is concerned more with the multidisciplinary nature and algorithmic perspective required to making progress in the field, whereas Trevor H.'s definition is, rather, a description in the developers' sense. Furthermore, the concept of automated learning of a model from data and generalizing on unseen examples, can be evaluated against a performance measure, and plays an important role: "A computer program is said to learn from

experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.", Tom Mitchell [34].

A Brief History and Future Prospects of Machine Learning is given in Appendix A.1.

2.2. Machine Learning Methods, Models and Algorithms

Machine learning is characterized not only by the intimate ties and overlaps with many related fields, but also by the diversity of ideas it has inherited from these fields, their modified methods and distinct principal goals. In ML, a computer program is able to draw conclusions by analyzing data and information autonomously. In the learning process, the agent does not require to be explicitly programmed. Conversely, it can change and improve its algorithm by learning from the result. The methods used by ML can be classified as supervised, unsupervised or reinforcement learning, based on the approach of training a model. These approaches are briefly described.

Supervised learning: Supervised learning is synonymous to learning with a teacher where the outcome of the model is known and the algorithm tries to classify data (i.e. solve classification problem) or predict outcomes (i.e. solve regression problem) as precisely as possible. These training or "supervising" algorithms use labeled datasets, both for inputs and outputs, so that the resulting ML model can measure its performance and learn over time.

Unsupervised Learning: Unsupervised learning methods do not use labeled data. They are usually employed for finding patterns among data points (i.e. clustering problem) or obtaining a low dimensional representation of data points such as the principal component analysis (PCA).

Reinforcement learning (RL): RL learns optimal strategy by sampling actions and then observing which one leads to the desired outcome. In contrast to the supervised approach, RL learns the optimal strategy not from a label but from a time-delayed "reward" signal that tells the system when it is doing well and when it is doing poorly. Over time, through experiments and trial and error, RL discovers a policy that will maximizes its expected reward. Thus, the goal of RL is to take actions that maximizes reward.

Although the terms **algorithms** and **models** are often used interchangeably, in the context of machine learning, they differ. Algorithms are procedures that are implemented in code and run on data– for example algorithms such as Logistic Regression, Decision Tree, Artificial Neural Network, k-Nearest Neighbors can perform pattern recognition, learn from data, and "fit" on a

dataset or a combination thereof. In this sense, ML algorithms have certain properties and features [32]:

- ML algorithms can be described with math (e.g. linear algebra) and pseudocode.
- The algorithm's "learning" efficiency can be analyzed and described, in relation to other specific algorithm.
- It is possible to use generic or devise entirely new algorithms such as through combining, different parameter configuration
- ML algorithms are implemented with modern programming languages such as Matlab and Python. It is also possible to create a library, such as scikit-learn library, through implementing multiple algorithms together and making them accessible with a standard application programming interface (API).

On the other hand, models are created by and are outputs of algorithms, and consists of model data and a prediction algorithm. For example, a ML algorithm in NNs result in a model comprised of a graph structure with vectors or matrices of weights having specific values. In almost all ML approaches, the role of the algorithm is to solve an optimization problem, by minimizing error of the model (i.e. data plus prediction algorithm) on the training dataset. The NN algorithm, for example, solves an optimization problem using back propagation and gradient descent algorithms to find a set of weights that minimize the sum (or mean) squared error on the training dataset. As a result, the resulting ML model has both data-i.e. vector of coefficients (weights) –and also a prediction algorithm- that multiply and sum coefficients with input data.

The said distinctions in ML together with its multi-disciplinary nature raises a challenge in terms of selecting algorithms and training of models while ensuring desired generalization. This has also been observed during the course of this dissertation work, and from two perspectives. Firstly, due to their optimization-centric approach, in which performance depends on the structure of the underlying problem, a high data volume and feature engineering is necessary to train a ML algorithm for complex task. Also, adequate output of the algorithm as a model require long training time, and have to generalize well before use. In this sense, the existence of large dataset and training time is presupposed. This has been a major challenge, as in many case the data and/or time of training is unavailable. Not fulfilling these requirements may lead to poor generalization which is often manifest as underfitting or overfitting of the ML model.

Furthermore, because the performance of ML algorithms are dependent on their parameters and the problem they intend to solve, an important difference between them and other models is their adaptability. Each algorithm group can be defined in the same way, but the input parameters and the specific task requirements can be different. Therefore, generalizations done by the ML model can differ, even when the learning approach and task objectives remain the same. Algorithms that are geared toward solving similar kinds of problems are available as a library in the form of API. In general, the user adapt algorithms in these APIs by customizing their parameter size, range and/or other implementation alternatives using heuristics and best practices. As a priori knowledge for "best performing" ML algorithm is not possible, using the mapping method may offer practical benefits to short-cut decisions for algorithm and algorithm parameter selection. This is probably the main reason that the approach has been applied in a broad range of simpler tasks that does not involve complex problems with bigger datasets. Nevertheless, it has to be kept in mind that this approach can only result in similar performances improvements and as such are not significant if simpler methods that are adaptable and scale well are desired. Techniques and methods to address these challenges are elaborated in Appendix A.2.

2.3. Modelling of Optimization Interfaces

In order to give a basis for the optimization interfaces which follow, it is helpful to consider the ML-based modelling problem from two perspectives. In the first, the modelling problem is defined as a process in which a model is trained to generalize form examples and make predictions on new samples. The learning process can be seen as a mapping of inputs to outputs, and hence as a function approximation. When this learning process is abstracted it consist of two algorithm layers-one that defines (describes) the structure and another that drives the dynamics. The structural description in ML is accomplished by defining attributes (parameters, variables) and functions. On the other hand, optimization algorithm drives the dynamics in ML in order to find values of parameters such as model coefficients through minimizing errors of mapping inputs to outputs. Therefore, in the first perspective optimization is considered as a common framework in the ML-based modelling problem [35].. Although there are advanced ML algorithms such as sequential quadratic programming (SQP), the gradient descent and its variant the stochastic gradient descent (SDG) optimization algorithms are found to be adequate for the tasks in this dissertation.

In the second perspective, optimization with machine learning is applied [36]. This, for example, is used in automated data pre-processing, hyperparameters tuning and model selection. In these problems, there is a non-linear objective function or design variables in large solution space. Consequently, bio-inspired and heuristic method such as particle swarm optimization (PSO) algorithm is used while optimizing with ML.

Environmental Model

One of the challenges of simulation based optimization is to reconstruct realistic solar radiation signals at different time scale [37]. The developed stochastic model is built up using connected procedures that consists of clustering, nonlinear regression and synthetic data generation.

i. Clustering for active dataset selection

Solar radiation values of typical days, which are representatives of the whole data sets, are chosen based on an optimum k-medoid clustering algorithm [38]. This algorithm results in a subset of the original data with cluster heads known as medoids. Given a data set P of N-dimensional entities $\mathrm{p_i} \in \mathrm{P}, \text{for i} = 1,2,\ldots,\mathrm{N}$, the k-medoids algorithm generates K non-empty disjoint clusters $\mathrm{S} = \{\mathrm{S1}, \mathrm{S2}, \ldots, \mathrm{SK}\}$ around the centroids $\mathrm{C} = \{\mathrm{c1}, \mathrm{c2}, \ldots, \mathrm{cK}\}$, by iteratively minimizing the sum of squared Euclidean distance

$$D(S, C) = \sum_{k=1}^{K} \sum_{p \in S_k} dist(p_i, C_k) \tag{2.1}$$

The Silhouette width, $\mathrm{S_W}$ is defined as

$$\mathrm{S_W}(p_i) = \frac{b(p_i) - a(p_i)}{\max\{a(p_i), b(p_i)\}} \tag{2.2a}$$

where $a(p_i)$ is the average dissimilarity of $p_i \in S_k$ to all other $p_j \in S_k$, $b(p_i)$ the minimum dissimilarity over all clusters, to which p_i is not assigned. The performance of the clustering is then measured as

$$S = \sum_{i \in P} \mathrm{S_W}(p_i) \tag{2.2b}$$

The minimum average distortion per dimension when fitting the K centers to the data is given by

$$d = \frac{1}{\dim(P)} \min_{C_1 \ldots C_K} E[(X - S_k)^2 \Gamma (X - S_k)^2] \quad (2.3)$$

Where Γ is the covariance of data X. Table 2.1 summarizes the steps of the optimal k-medoids algorithm. Since k-medoid clustering algorithm performs selection of representative days, the stopping criterion is iteration over all datasets. However, the number of cluster heads is set based on the average distortion per dimension when fitting these cluster centers to the data as given by Eq. 2.3. This process is illustrated in the case studies of section

Table 2.1. The Optimal k-medoids algorithm [38]

1	**Initialization**: set the optimal cluster size and S value as $S^{optimal}$ and $N_{optimla}$
2	**for** $k = 1, \ldots, M$ do
3.	Select k objects arbitrarily as medoid M_C Assign each remaining non − medoid object Nc to the cluster with the nearest representative Compute $S^k_{current}$
4.	For each pair M_C andN_C Compute the cost of swapping M with N as $S^k_{N \leftrightarrow M}$
5	Select N_C for which $S_{N \leftrightarrow M}$ is minimal
6.	If $S_{N \leftrightarrow M}$ is less than $S_{current}$ Swap N with M Set $S_{Current} = S_{N \leftrightarrow M}$ Go back to step 3
7.	else **calculate** S^k using Eqn. $2.2 - 2.3$ **If** $S^k > S^{optimal}$ **then** Set $N_{optimla} = k$ Go back to step 2
8	end for

ii. .Nonlinear Regression

Extending the clustering solution to the optimization task require mapping the output to be generalized using a parameterized functional form. To this end, a radial basis function network (RBFN) is used to approximate the solar radiation, $\hat{g}(x, \mathbf{w}) \approx g_\pi(x)$. The topology of RBFN showing the three main parameters for the connection weights, widths, and centers is depicted in Figure2.1. The ability to approximate any continuous function, compact structure and tolerance to noise are the distinguishing features of using RBFN [39], for regression of solar radiation.

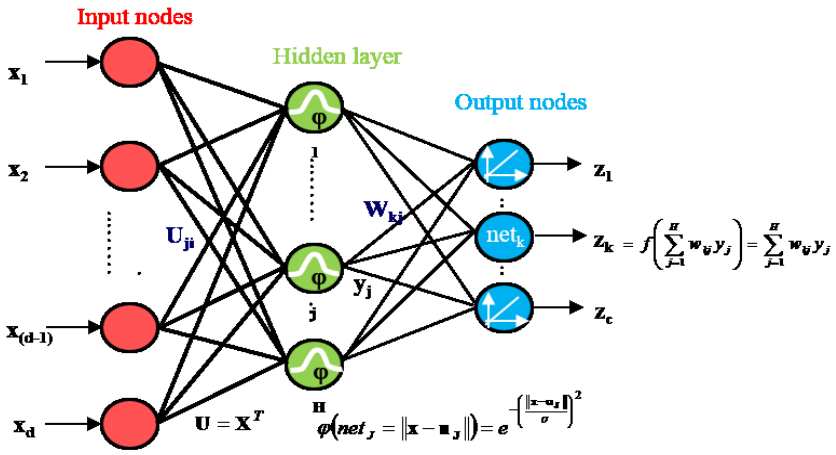


Figure 2.1. The radial basis function network (RBFN)

To refine the accuracy of approximation, for the most relevant states, a state distribution is defined such that:

$$\psi(x) \geq 0 \text{ and } \sum\psi(x) = 1 \tag{2.4}$$

$\psi(x)$ is defined to be the relative distance x from a cluster head, and is determined as:

$$\psi(x) = \frac{\phi(x)}{\sum_{\vec{x}}\ \phi(x')}, \forall x \in \mathcal{X} \tag{2.5}$$

where $\phi(x)$ is the variance of strength difference of state x and $\Sigma\phi(x')$ is the total solar radiation strength difference of all states. A cost function can now be defined as:

$$\overline{GE}(w) = \sum_{x \in X} \psi(x)[g_\pi(x) - \hat{g}(x, \mathbf{w})]^2 \tag{2.6}$$

Where $\hat{g}(x, \mathbf{w})$and $g_\pi(x)$ represent the estimated and measured solar radiation respectively.

The optimal weight vector w^* for which $\overline{GE}(w^*) \leq \overline{GE}(w)$ for all w, can be determined using SGD algorithm [40] as:

$$w_{k+1} = w_k + \alpha[g_\pi(x_k) - \hat{g}(x_k, w_k)]\nabla\hat{g}(x_k, w_k) \tag{2.7}$$

where w_{k+1} is the weight vector after the $(k+1)^{th}$ update, and α and ∇ are the learning rate and gradient operator, respectively. A larger α value is set at the initial training stage, which decrease dynamically as $k \to \infty, \alpha \to 0$ to ensure solution convergence.

Let each state x be represented as a feature vector:

$$\mathbf{v}(x) = [v_1(x), v_2(x), \ldots, v_d(x)] \tag{2.8}$$

that has the same length as w. Additionally, each component v_i of $\mathbf{v}(x)$ is a function known as a feature of x.

For the present work, states, x_1 and x_2 denoting two distance functions that measure compactness within and among clusters are defined. Each day's hourly solar radiation values are represented by these two states using variance of solar radiation strength difference. The cluster head with the highest members is first identified as the principal cluster head, and its hourly solar radiation strengths is identified. Afterwards, state x_1 is determined as strength difference of the remaining days' solar radiation values from their cluster heads. Similarly x_2 denotes strength difference, but between the cluster heads that represent a given day and the principal head. Finally, the problem of approximating each day's hourly solar radiation values is solved by designing a feature vector that captures both states and affine relationship and cluster interaction effects as:

$$\boldsymbol{v}(x) = [v_1(x) = x_1, v_2(x) = x_2, v_3(x) = x_1 + x_2, v_4(x) = x_1 x_2] \qquad (2.9)$$

PSO algorithm is used to determine the widths and centers of the hidden layer in RBFN.

iii. Synthetic Smooth Data Generation

Synthetic reconstruction of realistic solar radiation signals can be achieved using two approaches. The first one is by adding a fast signal component such as Gaussian white noise to the above mentioned, slowly varying signal. Using this approach, the solar radiation signal can be modelled by a differential equation that describes a stationary Gaussian process. Alternatively, it is possible to interpolate solar radiation values between intervals of measurements using efficient interpolators, for example, piecewise cubic Hermite interpolating polynomial (PCHIP) [41]. The second method is implemented in this work using MATLAB's built-in tool box.

Consumption Model

For developing the consumption model, a class of statistical models, based on demand related data such as daily production rates of industrial processes, are first constructed. Afterword, the operational parameters of the probability distribution for multiple load aggregates are reconstructed or learned concurrently using ML techniques. The industrial load aggregates could, for example, represent energy demands of processes connected to a heat distribution network.

There are many contributors and influences for stochastic demand in process industries and these depend on many factors. Variability of process temperature and time schedule influence this pattern, as well as manual operations of partially open industrial vessels. Furthermore, venting of steam through partly open steam valves (due to the absence of steam traps) can cause this variation.

In an attempt to reconstruct, while including these uncertainties, a stochastic model has been included to represent heat demand behaviors in process industries. The model is based on a discrete event and has two states (i.e. ON/OFF), which can triggers the on-off values, and manipulate the heat distribution network. In this scheme, a transition for ON (consumption) is fired according to an exponential distribution with mean μ_n, which depends on the daily production rates of industrial processes. To represent the OFF state transition, a random timeout duration based on a normal distribution with μ_D and σ_D is used. These values are initially chosen based on the survey carried out at the case study industry, and were tuned by a PSO algorithm afterwards.

Generic DER Model

A generic model is used to describe the variety of implementing distributed energy resource in an STS. The different DER technologies within which the STSs are realized (collector, storage, heat exchanger, etc.) are not relevant in this model. The DER model is abstracted through using a generic approach to the problem, without directly representing each kinds of technologies with all their constraints. This is valid since the ultimate functions of DERs is to convert to, store or convey thermal energy to industrial processes, supply networks or conventional plant. For these functions we have different DER technologies and different types of the same technology. However, all these DERs can be modelled similarly, because each DER has similar, but not identical, characteristics. The model can then be parameterized differently to represent heterogeneous DER models of STS in process industry. As uncertain the parameters are in these DERs, the RBFN is used to model their parameter. The implementation aspects of the RBFN is similar to that of the stochastic environmental model discussed above. The relevant DERs in STS that make up the generic model are given in Appendix B.

2.3.1. Reinforcement Learning Representation

Reinforcement learning is a goal-directed computational approach where a program learns to perform a task by interacting with an unknown dynamic environment. This learning approach enables the program to make a series of decisions to maximize the cumulative reward for the task without human intervention and without being explicitly programmed to achieve the task. The representation in RL can be abstracted through a constrained optimization approach using Markov decision process (MDP).

Constrained Optimization

Let x_t and u_t denote an n-and m-dimensional state and control vector and N_t be a stochastic process noise that describe a continuous time-varying and stochastic system as:

$$dx = f(t, x_t, u_t)dt + d\mathrm{N}_t \tag{2.10}$$

Further let J denote the objective function, comprising the time-varying cost $\Omega(t, x_t, u_t)$ and terminal cost $\Omega_T\left(x_{t_f}\right)$, which is minimized by finding the optimal control signal $u\left(t_i \to t_f\right)$ as:

$$J\left(t_i, x_i, u\left(t_i \to t_f\right)\right) = E\left\{\Omega_T\left(x_{t_f}\right) + \int_{t_i}^{t_f} \Omega(t, x_t, u_t)dt\right\} \tag{2.11}$$

Since we are dealing with a stochastic process, the optimization problem is to minimize the expectation $E\{\cdot\}$ of Eq. (2.11) over all stochastic trajectories that start in x_i as [42]:

$$J(t, x_t) = \min_{u(t \to t+dt)} \left(E\left\{\int_t^{t+dt} \Omega(t, x_t, u_t)dt + J(t + dt, x_{t+dt})\right\}\right), \forall x_t \tag{2.12}$$

The optimization problem formulation in Eq. (2.12) assumes that the final state at time t_f is known. However, such is not often the case in practice. To address this issue, a discount factor $0 < \gamma^t \leq 1$ is included in Eq. (2.12) to formulate a discounted infinite-horizon optimization problem as:

$$\begin{gathered} J\left(t_i, x_i, u(t_i \to \infty)\right) = \lim_{t_f \to \infty} E\left\{\Omega_T\left(x_{t_f}\right) + \int_{t_i}^{t_f} \Omega(t, x_t, u_t)dt\right\} \\ j(t_i, x_i, u_t) = E\left\{\int_{t_i}^{\infty} \gamma^t \Omega(t, x_t, u_t)dt\right\} \end{gathered} \tag{2.13}$$

Therefore, leading to:

$$J_\infty(x_t) = \min_{u_t \in U_t} \left(E\left\{ \int_t^\infty \gamma^t \Omega(t, x_t, u_t) dt \right\} \right), \forall x_t \tag{2.14}$$

Similarly, for the discrete time-varying stochastic system given by [43]:

$$x_{t+1} = f(t, x_t, u_t, w_t) \tag{2.15}$$

The discounted infinite-horizon optimization problem can be formulated as:

$$J_\infty(x_t) = \min_{u_t \in U_t} \left(E\left\{ \sum_{t=0}^{\infty} \gamma^t \Omega(t, x_t, u_t, w_t) \right\} \right), \forall x_t \tag{2.16}$$

Markov Decision Process (MDP)

Constrained optimal control implementing RL algorithm were developed based on, and has all the elements of, a Markov decision process (MDP) [43]. In MDP an agent, resides in an environment, which changes state randomly due to action taken by the agent. The state of the environment simultaneously affect both the immediate reward received by the agent and the probabilities of future state transitions. The objective of the agent is to learn to select actions that leads to the highest expected future reward. However, this kind of reward result in credit assignment problem: It is not possible to exactly determine which action had the deciding effect on the current outcome. To deal with this problem, the optimization problem is modified to maximize the expected sum of discounted rewards, which assumes that actions have exponentially decaying impact into the future. Thus, depending on the discount factor, it is possible to make the agent optimize for more short-term or more long-term reward.

The framework for the Markov decision process (MDP) is conceptually illustrated in Figure2.2 [43]. This framework allow abstraction of the learning dynamics for RL's decision making, process, which results from interactions between an agent and an environment. The agent is a computational entity (i..e RL algorithm) that can be viewed as operating in an environment and is able to continuously learn and make decision in order to meet specified objective(s) , which it tries to optimize. The environment, on the other hand, is defined as that part of the system within which the agent operates and cannot arbitrarily change but has an (external) influence upon it. From the perspective of dynamic system control, the controller logic takes the role of an agent and all that remains would make up the environment.

Figure 2.2. The general Markov decision process (MDP)

The constrained optimal control problem of dynamical systems, described by Eq. (2.10)-(2.16), has all the elements of MDP. At each time step the agent generate actions influencing the environment, then the states of the environment change. The aim of the interacting agent is to find the optimal policy that results in maximum reward measured in respect to the control objective. Basically, this policy is a mapping from the states to the control signals, and can be abstracted as an MDP, $\mathcal{M}$. MDP's abstraction of the learning process is characterized by the 5-tuple$(\mathcal{X}, \mathcal{U}, p(x', r \mid x, u), \gamma, R)$ comprised of [43]:

- $u \in \mathcal{U}$: action space of the reinforcement learning agent. In control, this is the bounded control inputs.
- $R \in \mathbb{R}$: expected reward from the system after the agent performs u at x. Rewards are generated based on a desired performance metric
- $p(x', r \mid x, u)$: is the probability of transitioning to x' and receiving r given $x \in \mathcal{X}, u \in \mathcal{U}$
- $p(x', r \mid x, u) \doteq \Pr\{X_t = x', R_t = r \mid X_{t-1} = x, U_{t-1} = u\}$ (2.17)
- Notice that p depends only on the immediate past, thus assumes that x_{t-1} and u_{t-1} contain information about the history. This property is known as the Markov property and is critical for successful RL applications in process control [44].
- γ: discount factor associated with future uncertainty, $(0 \leq \gamma \leq 1)$

Optimal Control Policies

The optimal solution to a reinforcement learning problem refers to the policy that generates the highest reward over a trajectory. Formally, an optimal policy must satisfy the principle of optimality: the optimal policy π^* is optimal if and only if $v_{\pi^*}(x) \geq v_{\pi+\pi*}(x)$ for all $x \in \mathcal{X}$ [43]. Note that there may exist many optimal policies (since we are dealing with probabilities in MDP), where $v_{\pi_i} = v_{\pi_2'} = \cdots = v_{\pi_N^*}$. The optimal value function is denoted mathematically as:

$$v^*(x) \doteq \max_{\pi} v_\pi(x), \forall x \in \mathcal{X} \tag{2.18}$$

Similarly, the optimal action-value function is denoted as:

$$q^*(x,u) \doteq \max_{\pi} q_\pi(x,u), \forall x,u \in \mathcal{X}, U \tag{2.19}$$

In a more explicit form, the optimal value function and action value function written in terms of Eq. (5.10) and (5.11) are given, respectively, by Sutton and Barto [43]:

$$v^*(x) = \max_{u} \mathbb{E}[R_{t+1} + \gamma v^*(X_{t+1}) \mid X_t = x, U_t = u] \tag{2.20a}$$

$$q^*(x,u) = \mathbb{E}\left[R_{l+1} + \gamma \max_{L_{t+1}} q^*(X_{t+1}, u_{t+1}) \mid X_t = x, U_t = u\right] \tag{2.20b}$$

Here, the max denotes that the optimal action will be taken in the next step and thereafter for the remaining of the trajectory. In theory, the optimal value functions can be explicitly solved for all systems using the above equation; however, such a task would require tremendous amounts of computation power even in trivial tasks.

Widely used methods for solving optimal policies in RL are dynamic programming (DP), Monte Carlo (MC) and temporal difference (TD). These methods can be further classified as model based and model free algorithms. In model based algorithm, the state transition probability has to be given explicitly. On the other hand, in model free algorithms a trial-and –error approach is used. In temporal difference (TD), the possibility for combining features from DP and MC enable learning system dynamics from interactions and immediate updating of value functions. These capabilities in TD allow development of simpler algorithms with relatively cheap computational cost, so it has been successfully applied in many of RL-based control tasks. TD uses techniques akin to unsupervised learning for predicting a variable's expected value in a sequence of states. Instead of estimating the total future reward, TD predict the joint reward that combine the immediate reward value and its own reward estimate at future time steps, whic is given by [43]:

$$\mathrm{G}_t = \sum_{i=0}^{n} \gamma^i R_{t+m+i} \tag{2.21}$$

In Eq. (2.21), the reward at time t is the combination of discounted rewards in the future, which means that the future rewards are valued less. The TD errors (δ), are the difference between the true reward value and the current prediction, and are given as:

$$\delta_{\mathrm{t}} = R_{t+1} + \gamma V(x_{t+1}) - V(x_t) \tag{2.22a}$$

$$\delta_t = R_{t+1} + \gamma Q(x_{t+1}, u_{t+1}) - Q(x_t, u_t) \tag{2.22b}$$

Similar to other optimization methods, the current values are added to the product of the learning rate and the errors in Eq. (2.22) while performing the update procedure as:

$$V(x_t) \leftarrow V(x_t) + \alpha[R_{t+1} + \gamma V(x_{t+1}) - V(x_l)] \tag{2.23a}$$

$$Q(x_t, u_t) \leftarrow Q(x_t, u_t) + \alpha[R_{t+1} + \gamma Q(x_{t+1}, u_{t+1}) - Q(x_t, u_t)] \tag{2.23b}$$

where ← denotes the update operator.

On convergence, the optimal policy is obtained using Eq. (2.23). Since TD methods are model-free, there is the issue of exploration vs. exploitation (which states that in order to learn action-values exploration is mandatory). To this end, the ϵ-greedy policy which allows exploring new options with probability $\epsilon \in [0,1]$ and performs the returns-maximizing action with probability$1 - \epsilon$. Additionally, ϵ is naturally decayed during the course of training and starts at a high value during the initial phase (when the agent has no knowledge). In due course, ϵ decays to a low value when training is nearly complete. There are two variants of TD methods that differ, only slightly, on the value update stages: SARSA and Q-learning [43]. In SARSA, the action-value functions are updated using Eq. (2.23) with the quintuple$(x_t, u_t, R_{t+1}, x_{t+1}, u_{t+1})$. On the other hand, only four parameters $(x_t, u_t, R_{t+1}, x_{t+1})$ are used during Q-learning updates (and assume u_{t+1} is a decision variable to maximize) as:

$$Q(x_t, u_t) \leftarrow Q(x_t, u_t) + \alpha \left[R_{t+1} + \gamma \max_{w_{t+1}} Q(x_{t+1}, u_{t+1}) - Q(x_t, u_t)\right] \tag{2.24}$$

SARSA is an on-policy algorithm meaning that its behavior policy is identical to its target policy. As such, it may quickly converge to a local optimum and never explore (since exploratory policies are typically not the optimal), resulting in a sub-optimal solution during training. In contrast Q-learning is an off-policy algorithm, which enable evaluation of several random policies with equal probabilities. In Q-learning it is possible to conduct deep exploration during training and switch to the optimal policy online. Thus Q-learning is guaranteed to find the optimal policy under the assumption that each state-action pair is visited infinite times and the probability of picking the optimal action under behavior policy is not nil.

Chapter 3

Design Optimization of Industrial Solar Thermal Systems

In an industry, the simultaneous presence of a range of both process and solar thermal technologies of similar types, would potentially result in several design variants. This may, in turn, lead to high uncertainties in the expected solar gains. Solar gain is the fraction of the load served by the STS generation. Furthermore, due to their changing dynamics, in which industrial heat demands vary with flow rate and temperature, the instantaneous solar gain as well as useful life of SIES show considerable variations. In addition, different control concepts are often implemented to allow a multi-mode of operation (e.g. summer/winter), and charge/discharge priorities in TES, leading to additional solar gain difference. Obviously, the energy cost competitiveness of the design variant in relation to the existing option should be viable. Addressing these issues in the design optimization of SIES requires the development of a performance criterion that will deliver maximum solar energy to the industrial process, minimize auxiliary energy and storage loss while meeting investment and operational constraints. Unfortunately, this kind of optimization problem formulation leads itself to a non-linear objective function that results in complex interaction of design variables in large solution space. This fact has forced many previous designs to be based on assumptions and restrictions that simplify the underlying complex process and objective function formulation. As a result STS of these designs lag behind their predicted performances.

In this chapter, an alternative design optimization scheme that leverage machine learning for design optimization of SIES is presented. This chapter is organized as follows: Section 2 presents the formulation of the ML-based design optimization problem. Section 3 discusses the layers challenges and limitations of existing modelling and simulation tools in design optimization of SIES. Section 3 discusses the need for heterogeneous and multi-scale modelling in design optimization of SIES, followed by Section 4, which presents the models used in this work. Formulation of a holistic design optimization is described finally in section 5.

3.1. Problem Formulation

There are many designs configurations and ways for joint resource and operation optimization of STSs in process industry, and these depend on many factors. Selection and sizing of solar collector technologies affect this process, as well as synergy with other boiler technologies or conventional industrial plant. The most pursued optimal sizing and placement of TES such as for adaptive modes of operation and flexible options of integration are some relevant examples. Like industrial processes, STS for its application has to be designed as a thermo-economic optimization of operations. However, this perspective is not yet considered in many of the existing design approaches. The solar-assisted industrial system is composed of a large number of interacting, nonlinear and uncertain subsystem, and therefore is complex. Reconstructing the dynamics of such a system is obviously not possible by only modeling each technology and industrial process separately, or by following a mono-approach modelling concept. To accurately represent and understand this system and optimize its design, it is not adequate to individually modelling of the same type of several technologies (and processes) and then parameterize their inputs and local condition similarly. It is necessary to incorporate additional capabilities of interactions, most importantly adaptive mode of operation and utilization flexibility in the design optimization.

For the design optimization problem, ML-based heterogeneous models encompassing stochastic and generic models are created. This allows disaggregated modelling of several similar technologies (and processes) and parameterizing of their inputs and local condition differently. It also enables representations of DERs' additional capabilities of interactions for optimizing ISTS's design. Evident from the case studies, these are essential features that cannot be offered by conventional methods,

3.1.1. Layers of Design Optimization

Two kinds of layers concerning design optimization are considered in this work.

Optimum Generation Profiles

Energy generation cost of STS is the determinant decision factor in its industrial integration, and depend on the optimal dimension of the solar field (SF). Use of representative cluster heads or their members (estimated from the RBNF model) enable construction of optimum generation

profiles which serve as a basis for the operation-based optimization that follows, as well as re-calibrate the SF size. These profiles enables continuous running of the operation-based simulation, whereby model evaluation is done within reasonable time horizons.

To this end, M sets of simulations each with N trials are performed, for generating random time series profiles from SF. The net thermal demand (according to the stochastic demand model), and the desired range of solar fraction, S_f are given as design specifications. For each trial, a random S_f value uniformly distributed between the desired ranges set the net contribution from the SF as:

$$\dot{Q}_{ns} = S_f \sum_{t=0}^{T_f} \sum_{i \in G_l} \dot{Q}_i^{ns}(t) \tag{3.1}$$

Where $\dot{Q}_i^{ns}$ is the thermal demands for in a regular daily cycle of T_f duration.

Time series of solar radiation (corresponding to the principal cluster head of the k-medoids algorithm 2.1) is used to find one possible optimal SF sizes so as to achieve these gains, by formulating the optimization problem:

$$\min_{A_C}\{\dot{Q}_{ns} - \sum_{t=0}^{T_f} \sum_{i \in G_r} (A_C I \eta_{sf} - \dot{Q}_{ls})\} \tag{3.2a}$$

The desired S_f is a constraint for avoiding oversizing of the SF and is set at design as:

$$S_{f,max} \leq S_f \leq S_{f,min} \tag{3.2b}$$

Where η_{sf} I and A_C represent the SF absorber efficiency, solar irradiation and effective total area of collectors respectively. $\dot{Q}_{ls}$ comprises thermal losses (piping and expansion vessel) as well as electrical parasitic (for tracking, start-up, shutdown and HTF pumping). In practice, the specific heat transfer coefficient (expansion vessel heat loss rate [kW] and piping heat loss/area [W/m^2] at specific temperature) are obtained from the manufacturer and/or are determined based on installation planning specifications such as dimension, material, insulation and pipe length. The SF absorber efficiency is determined from the generic model at the mean process/collector temperature. Note that the reaming M-1 profiles are generated by scaling the disturbances so they are comparable to the mean generation of the principal head (which is typical for fluctuations of solar radiations as given in Eqn. (3.2).

Operation Based Design Optimization

The energy generation cost also relates to the plants dynamic performance to improve its gain i.e. improve the plant operation duration having a constant quality energy output at the desired temperature. In this sense, operation-based design optimization such as for optimal integration point selection, control concept optimization as well as mode of operation and adaptive utilization strategies is required. This procedure leads to economic viability with options based on "subjectively" chosen industrial scenarios.

The heat distribution network can be equipped with TES capabilities in different points (Figure 2.1), both cascaded and distributed, in order to capture possible configurations. Furthermore, the vector of energies stored at the integration nodes are simulated, as well as the state of the storage (charging or discharging) and the state of the industrial process (switched ON or OFF). Simulations can be done for different scenarios, such as continuous operation without TES, or other more realistic scenarios including adaptive modes of operation and priorities of charging/discharging. By switching off an industrial process (through setting off in the stochastic demand model), the effect of demand side management (DSM) policy can be analyzed over different heat supply networks. Once sufficient demand data is collected under different scenarios, the design strategy of STS including TES integration can be optimized using this data.

For the scenario-based design optimization a high resolution data, in which the time axis is discretized uniformly as t=kδ with δ>0 representing time step is generated as discussed in the stochastic environmental model. The time step is chosen so as to adequately capture industrial load and ambient conditions changes. The on/off states of the heat exchangers and valves are determined at a slower timescale and hence are constant over k.

The design optimization process is carried out using different steps. First, the average thermodynamic activity of each node of TES integration are computed through minimizing a cost function without restricting storage capacities as:

$$\min_{\mathcal{G}_s(0),\ldots,\mathcal{G}_s(T_f-1)} \sum_{t=0}^{T_f} \mathrm{C}_{lm}\left(\dot{\mathrm{Q}}\left(\dot{\mathrm{Q}}^{s}_{G_s}(t)\right)\right) + C_{cp}\left(\dot{\mathrm{Q}}^{ns}\left(\dot{\mathrm{Q}}^{s}_{G_s}(t)\right)\right) + \mathrm{C}_{str}\left(\dot{Q}^{ns}\left(\dot{Q}^{s}_{G_s}(t)\right)\right) \quad (3.3)$$

Where $\dot{\mathbf{Q}}^{s}_{G_s}(t)$ represent the thermal power received by or drown from TES, and is given by

$$\dot{\mathbf{Q}}^{s}_{G_s}(t) = \frac{\mathbf{Q}_{GS}(t+1) - \mathbf{Q}_{GS}(t)}{\Delta} \quad (3.4)$$

C_{lm}, C_{cp}, C_{str} represent penalties for violating safe operation limits, violation of capacity constraints of auxiliary/conventional plants and loss or the thermal power drawn from TES respectively.

Then, the nodes that have significant impact on the cost function can be identified by adjusting the node removing factor, γ against the maximum TES capability in a heat distribution network. In a third step, the sets of nodes are re-adjusted against a different scenario using the reduced sets of nodes until the total TES power in a network showed significant increase. The operation based optimization scheme, for optimal control concept and TES integration is shown in Figure 3.1.

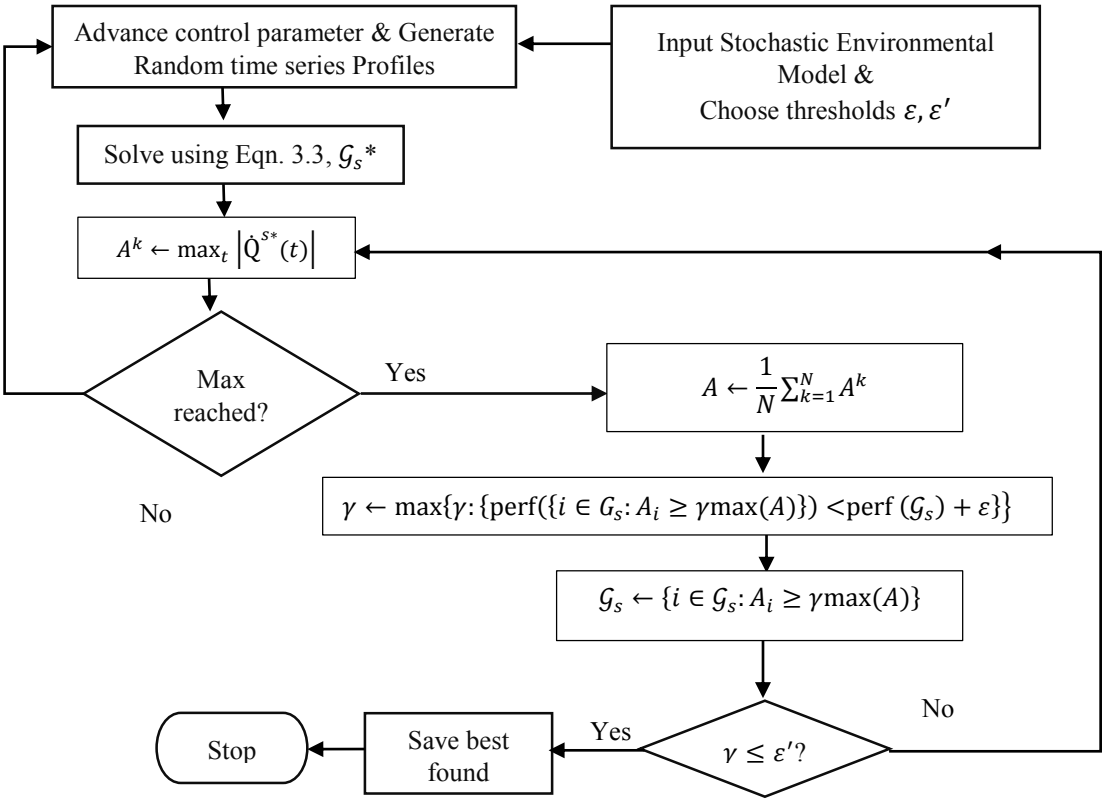


Figure 3.1. The operation –based design optimization flowchart.

Three modeling and simulation platforms namely TRNSYS [45], ANYLOGIC [46], and MATLAB [47] are used in this work. TRNSYS has been chosen due to its extensive library and modeling capabilities for renewable energy power generation. TRNSYS also allows interfacing to MATLAB and can link developed model to a generic optimization program GenOpt. On the other hand, ANYLOGIC computational platform is chosen because it is an object-oriented modeling framework suitable for complex systems that support system dynamic, discrete event as well as agent-based modelling and simulation. Moreover, ANYLOGIC has a metaheuristic

optimization engine based on the OptQuest package from OptTek Systems that allow parameter variation experiments, calibration of parameters, optimization, and Monte-Carlo runs. One objective of using machine learning is to integrate these platforms, or build two or more models for the same tool, that when interfaced, remove drawbacks present in either.

3.2. Demand Side Management (DSM)

Industrial loads are expected to play an active role in order to facilitate integration of fluctuating and periodically available generation from solar plant. In this context, industrial demand side management (DSM) improve energy efficiency as a result of minimizing loss through coordinated generation and consumption of power. DSM is achieved by enabling demand response through load shifting or shaving. Hence the effectiveness of DSM policy rests in the prediction accuracy of industrial loads and future energy generation capacity of the plant. The structure of the anticipated DSM strategy comprising the virtual solar sensor, optimized plant configuration, cluster classifier and DSM policy components is summarized in Figure3.2.

The solar radiation sensor component outputs the day-ahead hourly solar radiation DNI_{DAP} by predicting the future metrological context i.e. ambient temperature (using deep long-short-term memory (DLSTM) network) and the calibrated empirical solar radiation model (using deep reinforcement learning) as discussed in chapter four.

Next a cluster classifier takes in the predicted solar radiation vector and outputs the cluster group it belongs by first determining the two distance parameters discussed in the stochastic environmental model. Following the identification, time step energy generation profile of the optimal solar plant (associated with the cluster head) is used to generate a DSM policy. The DSM policy procedure starts by first calculating the energy factor,ζ which is a vector obtained from the ratio of the time-series predicted and identified cluster head's daily solar radiation value. The DSM policy is then generated by multiplying the design day's energy generation by the energy factor. The assumption here is that the energy generated from the optimal plant varies linearly with the solar radiation given a sufficiently small time step. This assumption is valid since the other active disturbances on the plant (ambient and inlet temperature) are effectively rejected by the implemented controller. Therefore, for the proposed DSM scheme, it is vital to ensure that the controller is rejecting temperature variation effects adequately.

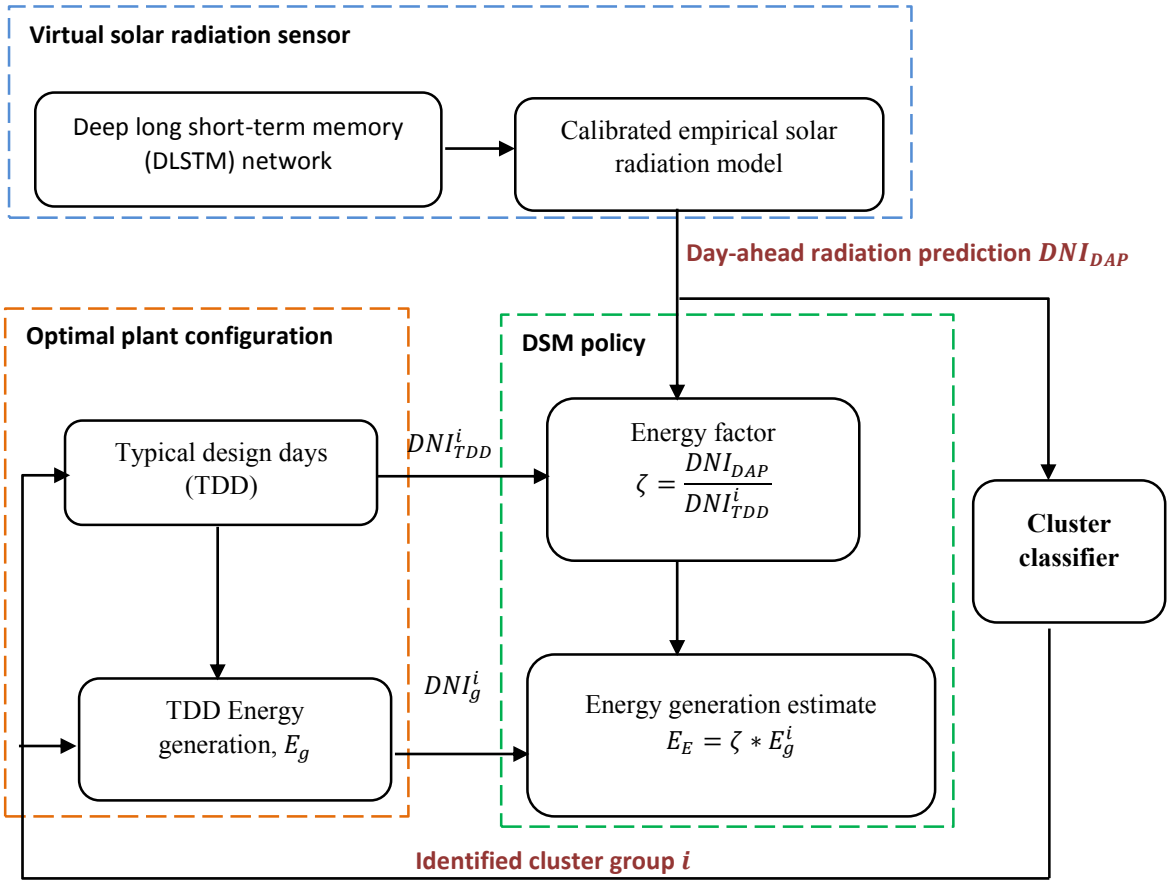


Figure 3.2. Schematics of the DSM strategy

Finally, the aforementioned design optimization approach will be elaborated once again through a case study as presented in the next sub-section.

3.3. Case Studies

Two case studies, one for solar process heating and another for hybrid solar-biomass cogeneration is presented. The investigated industry case study is found at Bahir Dar textile factory in northern Ethiopia (11°36′N 37°24′E). As with most regions in Ethiopia, the considered process industry receive high solar radiation, both global and direct, as shown in Figure 3.3.

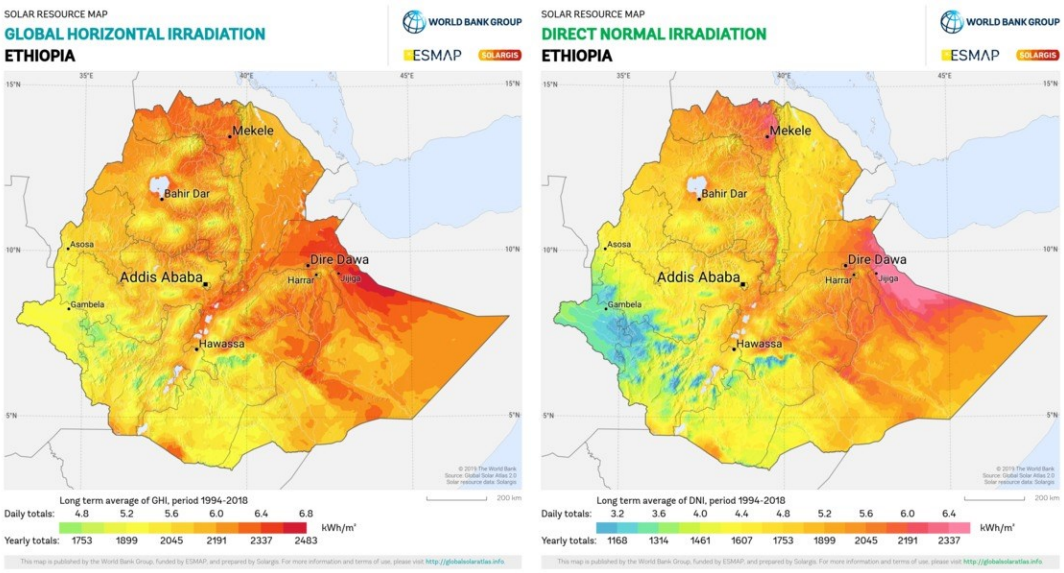


Figure 3.3. The chosen process industry (Bahir Dar Textile Factory) receiving an approximate global (left) and direct solar radiation (right) of about 6.0-6.4 kWh/m^2 and 5.2-5.6 kWh/m^2 per day [48].

Electricity, diesel and furnace oil (FO) are the major energy source to the plant with annual consumption of 27.54million kWh, 166kL and 615kL respectively. The textile factory heat demand are supplied by a steam network, which is fed centrally by two electrical steam boilers (capacity 3000 kW each). The boilers can each produce around 17-18 tons per day saturated steam at 5 bar and 152 °C. The steam demand depend on the operation of wet processing machines such as bleaching, de-sizing, washing dryer, calendar, jigger, sizing, etc., Although, heat is used indirectly via internal exchangers, direct steam injection, due to its efficiency to transport thermal energy, is mostly used to heat industrial process vessels at low temperature. In addition to the electrical boiler, the textile plant has installed FO fired steam boiler for a secondary dyeing machine. This boiler operates only during production of dying section and consumes an average of 250 litters FO per day running with efficiency below 65%. The average annual FO consumption of the boiler is 65,923 litters. Increased price and poor reliability of FO are the two main problems that challenge the dyeing section of the textile factory. Data for the case study are collected from the factory and some are obtained from Ethiopian Electricity Agency (EEA) that conducted energy audit in the factory for its "Energy Efficiency Regulatory Framework Development and Implementation" work. In addition, solar radiation data are collected from the research center at Bahir Dar institute of technology located at close vicinity to textile factory. A minute interval solar data are logged from a crystalline silicon sensor and Kipp & Zonen CM 11

pyranometer enabled by a Vaisala WXT520 weather station. The preprocessed one year solar radiation data, for a 15 min model time, consist of 35, 040 data points.

3.3.1. Process Level STS Integration

Direct steam injection, as in many process indsutries, is used to heat the dyeing process vessel in the textile factory. The discharged steam then condenses and relinquishes its heat to the surrounding liquid to raise its temperature (maximum temperature level is about 90 °C). The jigger dyeing machine's heat requirement depends on several factors such as type of dye, liquor to material ratio, processing time, batch capacity, and steam supply pressure. In order to represent steam consumption, a discrete event having two states (ON/OFF) is implemented. A transition for ON (consumption from steam distribution) is fired according to an exponential distribution with mean μ_n, which depends on the daily production rates of the wet processing machine. To represent the OFF state transition a random timeout duration based on a normal distribution with μ_D and σ_D is used. These values are initially chosen based on the survey carried out at the textile factory, and were tuned by PSO algorithm afterwards. Figure3.4 depicts the stochastic model output, which was calibrated based on a two days measurement on the factory's electrical boiler and the previous work of the author [49].

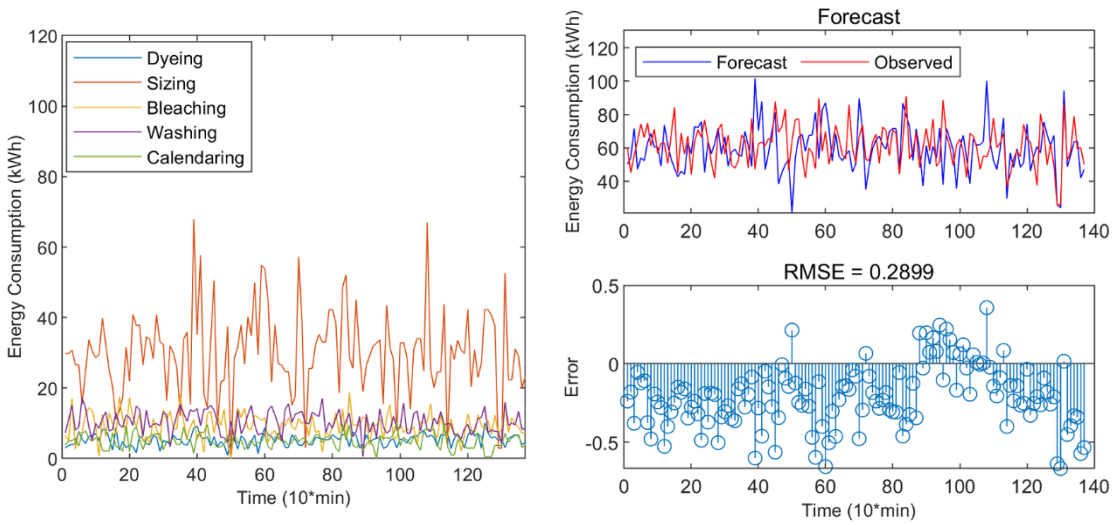


Figure 3.4. Stochastic heat demand patterns representing (a) individual wet processes and (b) model error of aggregated consumption of processes connected to a steam distribution network.

There are three sub-process in each dyeing stages (dyeing, soap at boil and after treatment). Based on the stocastic demand model and relevant machine‘s specification, the demand profile of the

dyeing process is identified (Table 3.1). Table 3.1 also depicts that except the wash at boil the remaining processes operate at temperature between 45 °C and 75 °C. Hence, low temperature solar collector can augment the heat demand in the dyeing process and substitute the existing inefficient (efficiency around 63%) FO steam boiler.

Table 3.1. Jigger dyeing machine key specifications

Process type	Cyclic, batch process
Batch fabric size, kg	50-70
Average daily production capacity, m2	4,900
Machine efficiency, %	75
Minimum fabric weight, kg/m2	0.265
Dye type	Reactive
Liquor to material ration	5
Batch sub-processes	Dyeing, wash at boil and after treatment
Sub-process temperature, °C	75, 90, and 50
Sub-process heat demand per batch, Kwh	25, 45.3, and 13.6
Process time, h	1,.3-.45,.15-.25

The solar collector technology used for process level integration is Apricus APSE-30 ETC , and is listed in the Solar keymark database for certified products [50]. Table A.2 shows the performance parameters of the chosen ETC. On the other hand, the cyclical batch process, with sub-processes, is represented in Anylogic using states. States represent a location of control that is able to respond to external triggers or events. The schematics of STS integration and modeling of the daily dyeing process is shown in Figure 3.5.

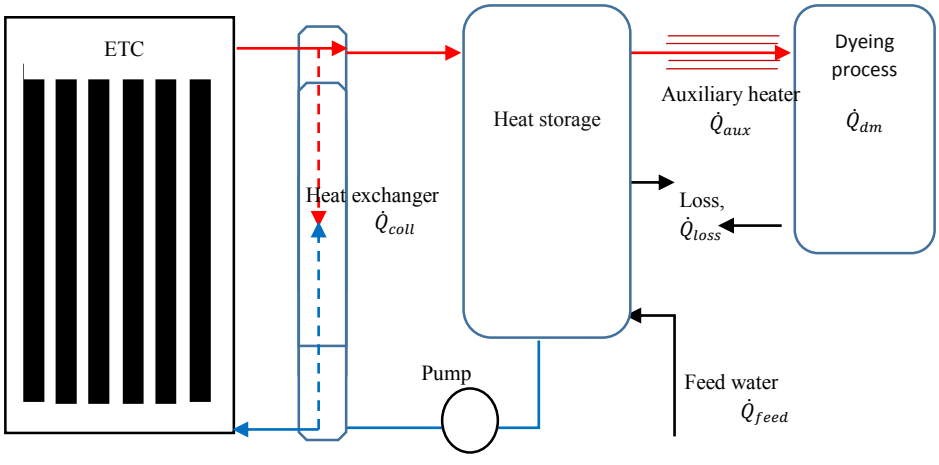


Figure 3.5. Schematic diagram of the proposed solar system with auxiliary heat for the dyeing process (left) and modelling of the batch process consisting sub-process with their associated transition (right).

A solar fraction range of 40-70%, according to Eqn.(3.1) is used to generate a random time series profiles from the solar field. The environmental model (as elaborated in section 2.1), composed of three connected processes of clustering, nonlinear regression and synthetic data generation, is then used. The said section explains the procedures for selection of representative cluster heads as well as their representations as Gaussian distributions. Furthermore, each day's representation as states defined by two distance functions is also discussed. Accordingly, 4-medoids cluster centers explaining 65% of the annual data variance are chosen arbitrarily. Afterwards, the two distance states that are inputs to the RBFN model are also determined. The performance of the clustering algorithm and features of the transformed data points are shown in Figure 3.6.

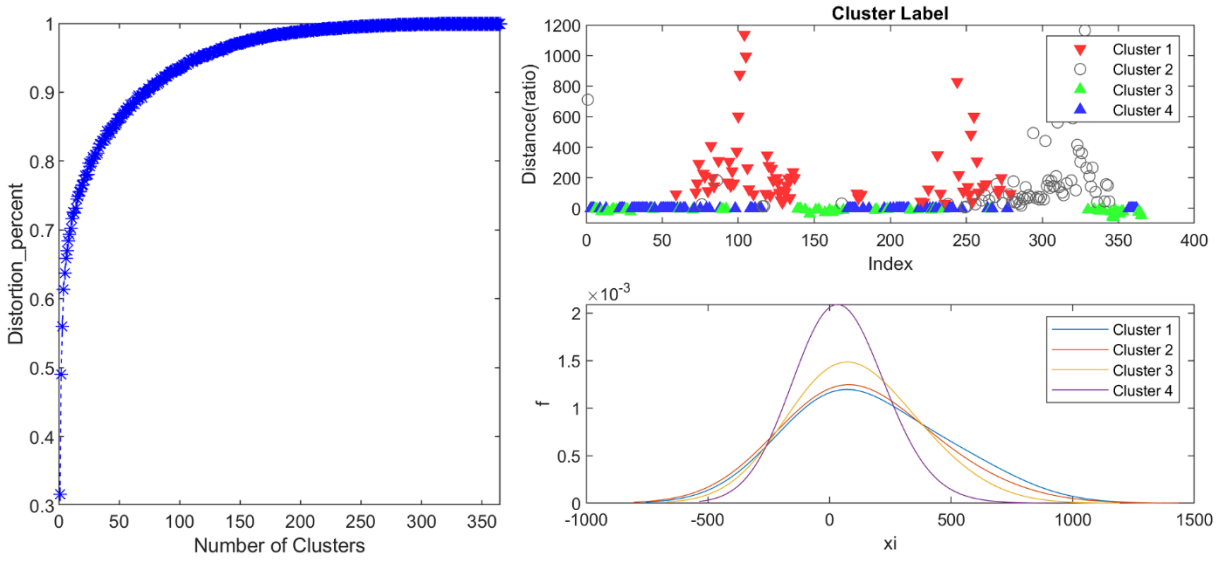


Figure3.6. Trend of cluster size VS variance of data points (left) as well as the stochastic environmental model output showing each day's representation as states defined by two distance functions and typical design days' representations as Gaussian distributions**.**

The operatiion-based optimization problem is formulated as:

$$\Omega = \sum_{0}^{T_F} \delta_1 \|Q(t)_t - Q(t)_{dm}\|^2 + \delta_2 \|Q(t)_{aux}\|^2 \tag{3.5}$$

Where the energy dynamics $Q(t)$ in the system (Figure3.5) depend on the solar heat gain from the evacuated collector, $\dot{Q}_c$, the auxiliary heat supply, $\dot{Q}_{aux}$, feed water enthalpy, $\dot{Q}_{feed}$, the system losses, $\dot{Q}_{loss}$, and the heat demand of the dyeing process, $\dot{Q}_{dm}$, and is given as:

$$\frac{dQ}{dt} = \dot{Q}_c + \dot{Q}_{aux} + \dot{Q}_{feed} - \dot{Q}_{loss} - \dot{Q}_{dm} \tag{3.6}$$

It is possible to give different weight factors ($\delta_1$andδ_2) to the supply-demand mistmatch as well as to the auxiliary heat variation. In this work the weight factors are given the same values. During the optimization three control concepts (CCs) are considered.

CCI: This is a fixed flow rate, on/off control:

$$u[k] = \begin{matrix} \dot{m}_{ON} & if\ T_{str,i} - T_{str,o} > S_d \\ 0 & otherwise \end{matrix} \tag{3.7}$$

This method is currently used in most solar controllers. The controller monitor the solar thermal systems through a relay actuator that turn the solar pump on and off. In CMI the flow rate is not varied but is simply turned on or off according to differential temperature between storage inlet and outlet (or sometimes according to the fluid temperature at the collector outlet).

CCII: Emulates a PID controller as

$$u[k] = u[k-1] + 5\,3.7667e[k] - 6.333e[k-1] + 2.6667e[k-2] \tag{3.8}$$

The PID controller manipulates the heat exchanger inlet flow from storage so that the heat exchanger temperature output in the storage loop tracks the a constant reference temperature set point under the presence of disturbances (such as solar radiation, storage outlet temperature variation and ambient temperature change on collector and heat exchanger network).

CCIII: A daptive control based on solar radation tracking as

$$u[k] = \frac{\dot{m}_{max}}{2\pi}\,(G^*[k-1] - \sin(2\pi G^*[k-1]) \tag{3.9}$$

Where $\dot{m}_{max}$and G^* are flow rate and global solar radaation scaled by the daily maximum value.

The collector return temperature dynamics under a daily solar radation varation (principal cluster head) and the above control concepts is shown in Figure 3.7. CCI shows undesirable temeprature oscillation, especially when the change in solar power is high around the solar noon. This leads to excessive heat loss and hence low thermal system performance. On the other hand, a simulation based optimization under CCII and CCIII conducted on the same representative day as of CCI, is shown to rectify this problem. In particular CCIII allow enhanced STS performance, through adequate manipulation of flow rate according to the solar radiation cahnge. These performance differences of the control concepts can be verified by comparing their uself solar gains, depicted

in Figure 3.8. From this figure, CCIII leads to higher overall solar gain, while CCII shows improved performance even during low solar radation months.

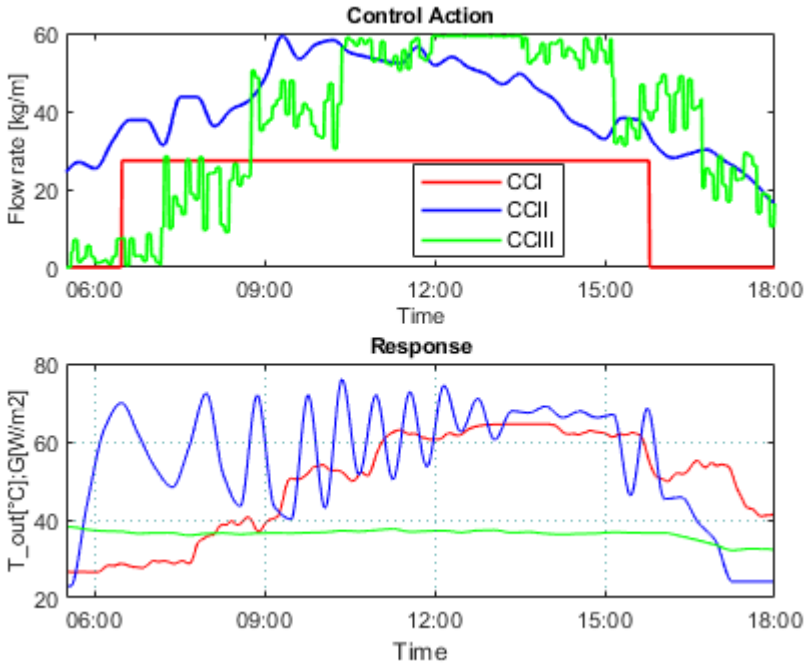


Figure 3.7. The collector return temperature dynamics under different scenarios.

As outlined in the previous section, Figure 3.8 indicate that further solar gain improvement through DSM is possible. However, it is important to note that unlike the adaptive control concept (CCIII), the PID controller is tightly tuned around a certain operating point (and has fixed parameters as shown in Eq. 3.8). Therfore, although the PID controller performes well when the solar radiation variation is within the designed operating point (November to February), its performance degrades nonetheless as influences change.

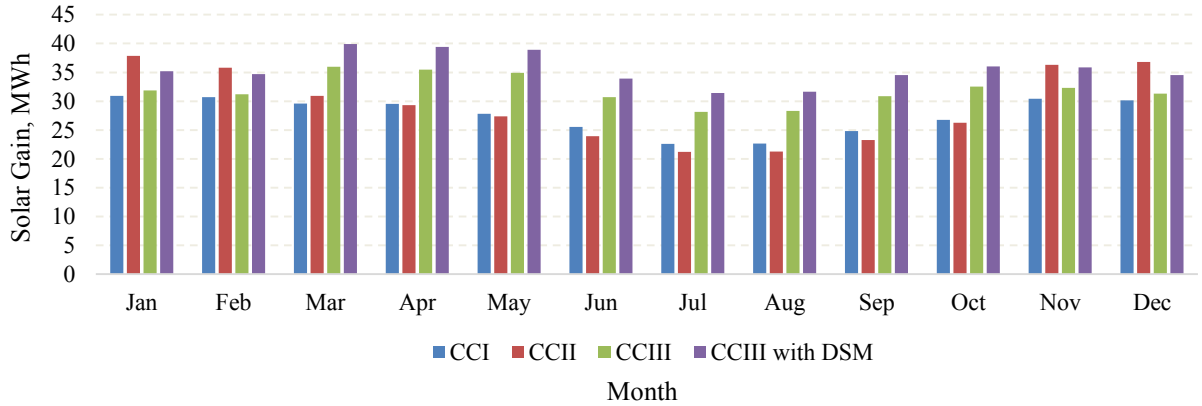


Figure 3.7. Solar gain Performance comparison between CMI and CMII

Finally, the adequacy of the stochastic model for optimization task is investigated. Performance of the identified four cluster heads, with and without the use of the RBFN model, on representing the net solar gain shown in Figure 3.8. As depicted, the representaive design days can be used, without compromising the design optimization task. Furthermore, their performance can be improved by coupling them to the RBFN model.

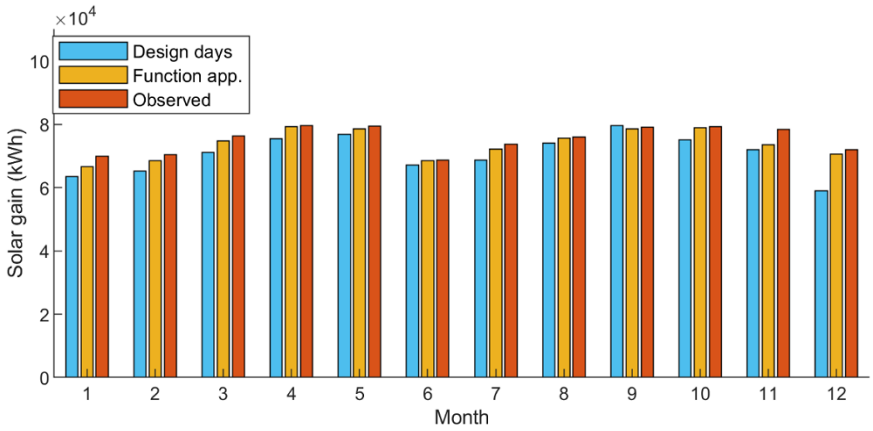


Figure 3.8. Effect of clustering and function approximation technique on the net solar gain.

Finally, Table 3.2. establish the feasibility of the STS designs for the dying idutrial process. Due to the abscence of valid data, the price of all controllers is not considered in the investment cost.

Table 3.2. Key feasibility indicators of the ISTS under electricity cost uncertainity.

Indicators	@ 0.0192 €/kWh		@ 0.03 €/kWh	
	CCII	CCIII	CCII	CCIII
Solar gain, kWh	341.35	383.72	341.35	383.72
Solar fraction, %	56.28	63.26	56.28	63.26
Investment cost, €	175, 584	175, 584	175, 584	175, 584
Payback period, yr.	5.7	5.4	6.9	6.3
Annualized total cost, €/yr.	32,792	30,881	39,521	36, 535
Co_2 reduction, Tone	252.18	251.20	252.18	251.2

3.3.2. Solar-biomass Cogeneration Plant

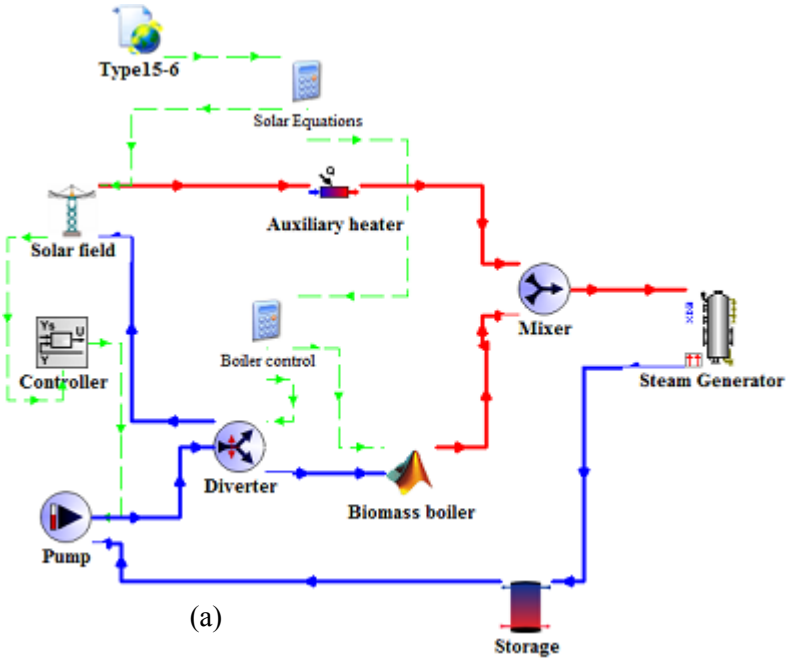
As with the process level, the ML-based method can be used for feasible design solution of STS to conventional plant integration. Consider the optimal design problem of the hybrid solar-biomass cogeneration plant depicted in Figure 3.9 [51]. In this design, a molten salt biomass boiler is modeled in MATLAB and integrated to a solar plant model in TRNSYS. The model in TRNSYS does not support the heat transfer fluid (HTF) used in this work. The resulting

configuration is latter optimized in GenOpt to minimize biomass power utilization index (BPUI) and excess saturated steam generation (ESSG). The hybrid plant comprises two parts: The first part, illustrated in Figure 3.9(a), shows the saturated steam generation (SSG) from the solar field and biomass boiler. Whereas the second part (Figure 3.9(b)), shows the superheating and power generation (SPG) aspect of the hybrid plant.

The main aim of the work is to carry out operation-based design optimization, to simultaneously address both dynamic performance improvement and sustainable energy supply to the case study process industry. The dynamic performance improvement aspect of the design optimization rely on operational scheme that guarantee constant quality energy output at desired temperature (desired dynamic performance) at the input to the heat recovery steam generator (HRSG), despite solar radiation and industrial demand variation (Figure 3.9(a)). To this end, alternate operation of the solar field and biomass boiler is implemented as follows: when the useful solar beam radiation is above a certain level (and is known to fluctuate), vary HTF flow rate and attain desired performance at HRSG; on the other hand, when the solar radiation is not available operate a biomass boiler and vary the HTF temperature while keeping its flow rate constant.

The aforementioned alternate operation of the solar field and biomass boiler is enabled by cooperative control of a pump (enabling variable/constant HTF flow rate) and a two-way diverter (conveying HTF flow through the solar field and/or biomass boiler). These two controllers are designated as “Controller” and “Boiler control” in the schematic depiction of the hybrid plant (Fig 3.9(a)) and are connected to the “Solar Equations” to get the solar beam radiation input. In addition to their usual control task of regulating the HTF temperature (through manipulation of flow of HTF or combusted biomass), these controllers have a threshold detector logic that is flagged on/off whenever the useful solar beam radiation is larger/lower than a set value. As a result, the “boiler control” set the flow portion through the two-way diverter and also regulate biomass combustion (including start/stop operation of the biomass boiler). On the other hand, the solar field “Controller” switch on/off the pump mode of operation (resulting in variable/constant flow rate of the HTF), and also perform regulation of the HTF temperature while the pump is in the variable speed mode. In this sense, the implemented control scheme is an alternating continuous control of flow rate (of HTF/combusted biomass).

Consequently, when the useful solar beam radiation is larger than a set value, the mass flow rate of the HTF through the pump varies in order to keep the solar field outlet temperature at the desired level i.e. around 450ºC. In contrast, when there is low useful solar radiation and the biomass boiler is turned on, the mass flow rate of the HTF through the pump is kept constant and negligible energy is taken from the solar plant. However, although the HTF flow rate is constant, the biomass boiler will be on, and HTF temperature can still be made to vary due to the heat exchanged from the combusted flue gas. This way, the required saturated steam energy at the HRSG, due to flow from either the solar field or biomass boiler branch of the two-way diverter (Figure 3.9(a)), is maintained at all times regardless of variable/constant HTF flow rate. The modelling of the individual parts comprising the hybrid solar/biomass cogeneration plant can be referred from the author's work [51].



(a)

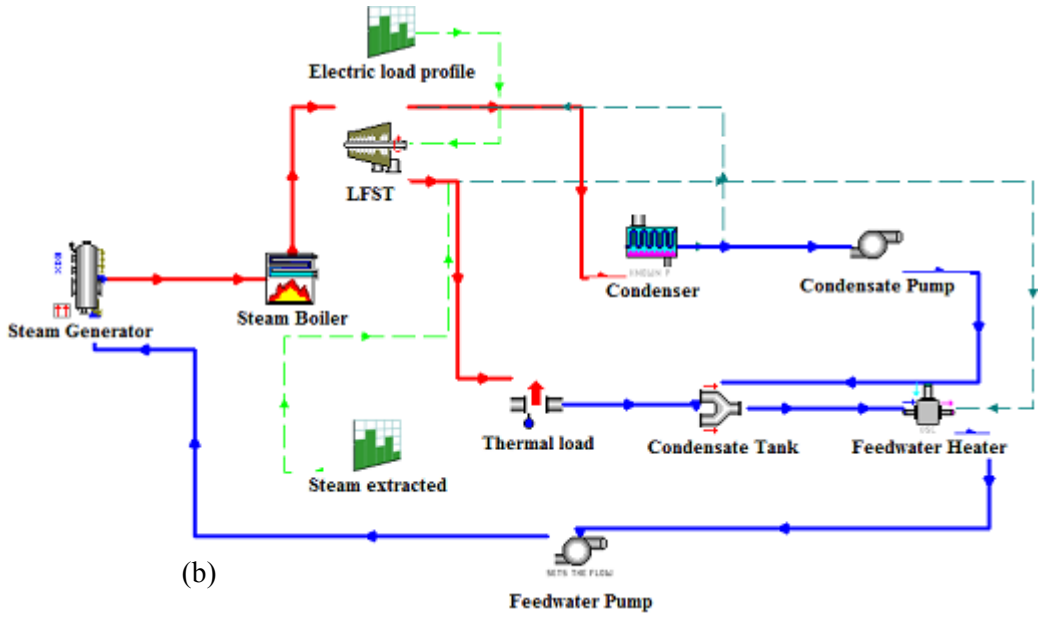


Figure 3.9. Graphical representation of the hybrid solar-biomass power plant modelled in MATLAB/TRNSYS showing (a) the saturated steam generation, and (b) the super-heated steam and Rankine power cycle.

The design optimization of the hybrid solar plant starts with this considerations: the primary energy source of a biomass plant is controllable while that of a solar plant is not. Thus it is possible to realize economical and sustainable energy supply from the hybrid plant to industry through lowering the biomass power utilization index, $BPUI = \eta_b^{gain}/\eta_{sol}^{gain}$ while avoiding oversizing of the solar field and/or excess auxiliary heat by minimizing net energy balance of the SSGC. The excess saturated steam generated (ESSG) by the solar-biomass thermal system based on the system boundary presented in Figure 3.9(a) can be expressed as:

$$\frac{dQ}{dt} = \dot{Q}_{sol} + \dot{Q}_{bb} + \dot{Q}_{aux} - \dot{Q}_{HRSG} - \dot{Q}_l \tag{3.10}$$

Therefore, the design optimization of the hybrid solar-biomass plant is formulated as follows. Given time-varying industrial thermal and power demand as well as fluctuating solar radiation, the problem is to determine the optimum size of the solar field and storage tank to minimize the objective function as stated in Eq. (3.11). It is possible to give different weight factors (δ_1 and δ_2) depending on the biomass availability and investment constraint. For the optimization procedure the developed model is linked to the generic optimization program GenOpt.

$$\Omega = \left[(\delta_1 \mathrm{BPUE}) + \delta_2 \sum_{k=p}^{q} \dot{Q} \right] \tag{3.11}$$

The capability of the aforementioned design optimization method, leading to a feasible energy solution can be demonstrated from the depiction of Figure 6.9 and 6.10. To this end, low allowable solar radiation value (during the first week of August and October) is used. The design solution ensures that the operating temperature of the HTF (is between 160 ºC and 280 ºC), and is within the safe margin (200 ºC to 450 ºC). Moreover, the inlet temperature of HTF to the solar field, for both cases, is greater than its freezing temperature (about 145 ºC). This condition is an important requirement for the operation of the solar system. According to Figure 3.10, the HTF inlet temperature. Figure 3.10 illustrates the performance of the control concept in keeping the outlet temperature of the solar field at the set point value. As explained, the useful solar beam radiation enable the controller to work only during the daytime.

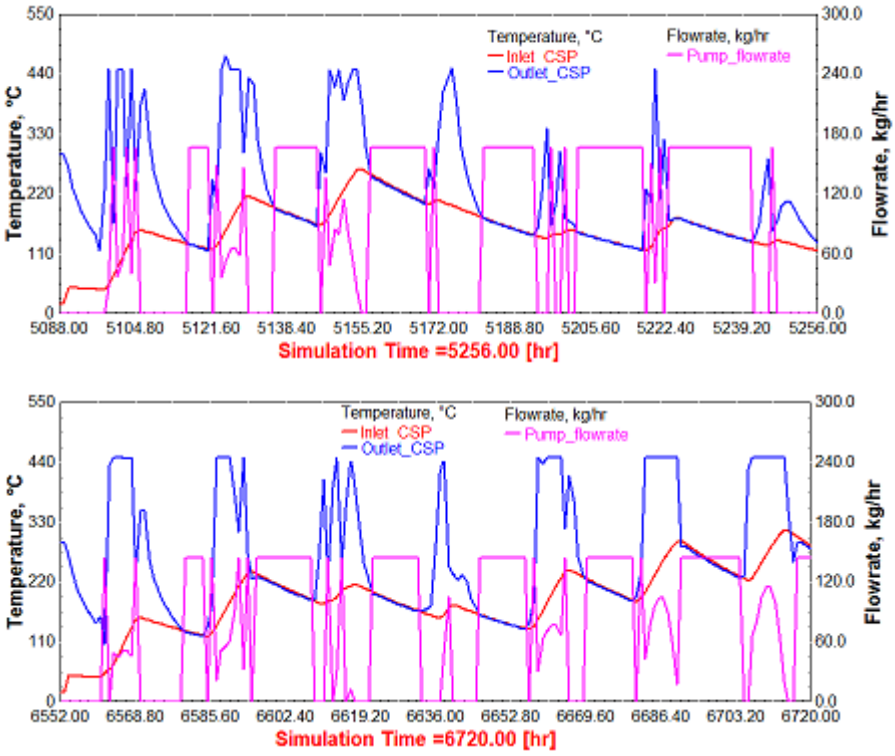


Figure3.10. Solar field inlet and outlet temperature with the associated HTF flow rate for one representative week in the month of (a) August, and (b) October

Similarly the dynamics of individual thermal components along the source and load side of the HRSG allow efficient operation of the hybrid plant. It also shows the steam flow rate to the gas boiler (Figure 3.11).

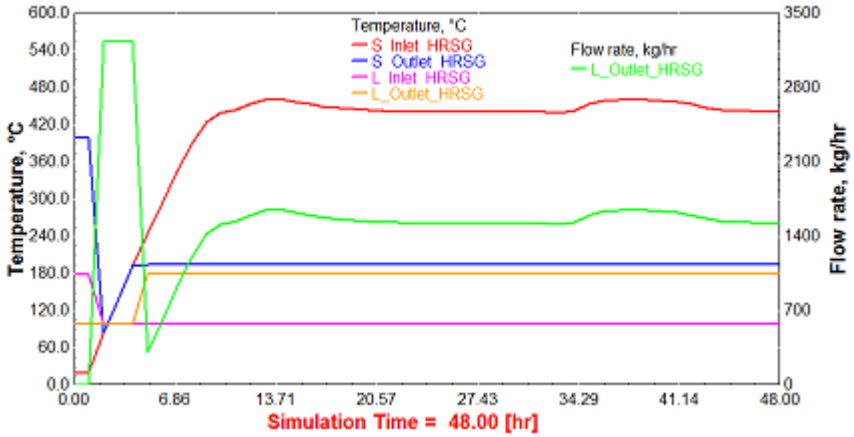


Figure3.11. Temperature dynamics and flow rate in heat recovery steam generator (HRSG).

Overall, the optimal design leads to a plant efficiency of 31.5% with 23.5% solar gain resulting in about 0.094$/kWh levelized cost of generation. Furthermore, considering global power outage loss, the hybrid plant could be seen as a preferred industrial source of energy.

Chapter 4

Deep Reinforcement Learning based Constrained Optimal Control

One prominent scientific and technological challenge encountered nowadays is to optimally control physical systems whose underling processes are complex for accurate modeling. Complexity in such systems relates to the multi-input multi-output (MIMO) relations having changing dynamics, nonlinearities and uncertainties. Systems complexity results in their inaccurate modeling leading to optimization and/or control difficulty. In such cases, the physical systems operate at suboptimal states or are expensively calibrated by domain experts [52].

One such example is robust control of STSs in process industry. These systems have nonlinear dynamics, and are not time invariant due to shift in operating points. As a result, adequate control of such systems is difficult. The difficulty is due to the requirements for their smooth integration as well as optimal operation while fulfilling design and demand constraints [53].

To overcome the limits, reinforcement learning (RL) based constrained optimal control schemes, have been introduced as a potential solution [54], and continue to result in improved control performances compared to conventional methods [44]. This achievement is due to advancement in processing technology that enable faster model evaluation as well as the availability of accessible development frameworks and efficient metadata techniques for model training. RL-based optimal control schemes offer to main two advantages over conventional methods. The first one is the possibility for a derived model-based control as well as online optimization of processes with the ability of prediction of system behavior [55]. The second advantage comes from the realization of a virtual sensor that can replace physical sensor for reasons related to cost or unreliable operation [56]. This kind of virtual sensor can provide indirect measurement of signals as well as serve as a fallback solution.

In this research work both of the aforementioned capability are explored. The rest of this chapter is organized as follows. In Section 2 the role of deep NN models for RL-based optimal control is

discussed. This is intended to show features and properties that are relevant to optimal control problem. Taking into consideration these factors, the implemented deep deterministic policy gradient (DDPG) algorithm is presented in Section 3. Section 6 summarizes key benefits and challenges in deep RL-based optimal control. This section also enumerates remedies identified and implemented during the course of this PhD work. Finally, a case study elaborating the detailed design, training process and comparative performance of the DDPG algorithm is demonstrated in simulation.

4.1. Deep Reinforcement Learning

RL control, as discussed in chapter 2, using classical methods such as Q-learning has implementation limitations, most notably the "exploration/exploitation dilemma". This means that the particular state-action sequences the agent observes are controlled by its own decisions, yet it does not discriminate which regions are more productive than others, resulting in slow convergence or bad solution. In addition, the state-action transitions needs to be fully specified as an input or their probabilities discovered by the slowly converging algorithms. These issues has raised concerns for the practical applicability of the classical RL approach for real-world problems. Although the classic RL algorithms can work well be on small, discrete state spaces, the dimensions of the state and action spaces increase exponentially for high-dimensional problems, which makes the computation of the optimal strategy intractable. To handle this challenge, function approximation, without directly learning the expected outcome of taking an action in a given state, approximate it through regression techniques. The advantages of such scheme, besides reducing training iterations for getting good state-action pairs, enables distinguishing productive state-action space regions based on similarity between states. Particularly, deep neural network (DNN) as the function approximation schemes in RL continue to result in breakthroughs across multiple fields, including computer games, autonomous driving, and optimal control. These breakthroughs can be primarily attributed to the nature of these networks to be trained online (i.e., one training example at a time), and allowing their update after each action.

Figure 4.1 depicts the influence of deep RL worldwide from 2010 – 2020 based on the growth trend in search result on Google. From this figure, it is possible to see that the interest in RL stayed somehow stable until 2015, which marked the start of significant growth. According to

[44], the main contributor that triggered the sustained growth was the pioneer work by Mnih et al. [58] that developed a deep RL algorithm, called deep Q-learning network (DQN). In that work, authors used the DQN algorithm to learn the optimal play actions for many Atari games from image inputs alone, and using the same algorithm for re-training an agent for each game. Performance of the DQN agent surpassed human level in half of the games. Inspired by it, deep RL, combining Q-learning with deep neural network (DNN) has been successfully applied to enhance the performance of complex control tasks.

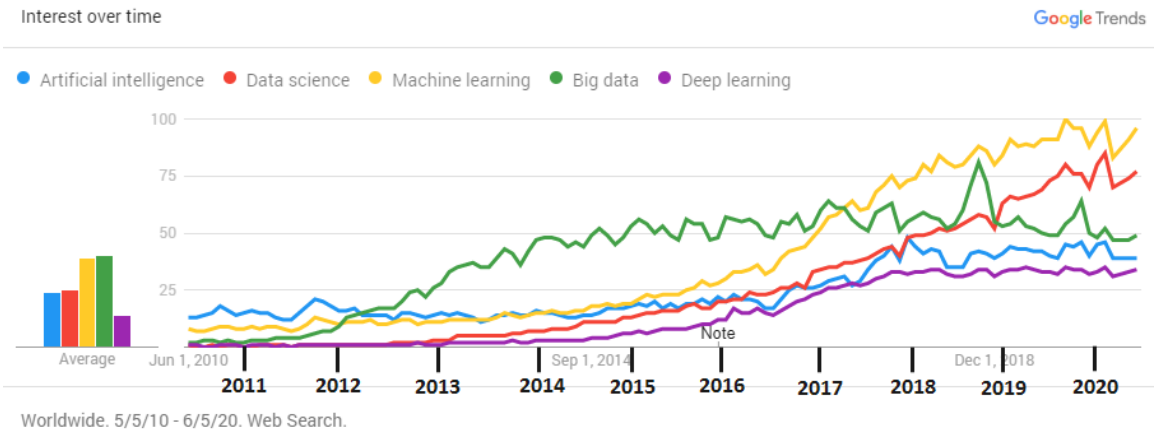


Figure4.1. Worldwide web search on application areas of machine learning between 2010 and 2020.

Unfortunately, although DQN achieve model-free optimal control, it cannot be directly applied to process control that require continuous actions. Therefore, DQN, at that time, was adapted to process control by discretizing the action space. Regretfully, this approach has resulted in major drawbacks. One of the major drawbacks was the curse of dimensionality: agents that represent multi-dimensional and continuous control tasks, require a large number of states and actions, which makes the computation of the optimal policy intractable. In addition, traditional DQN methods are not able to handle operational constraints systemically and adequately. In order to address the aforementioned issues, a deterministic policy gradient (DPG) algorithm, a policy iteration algorithm, was developed using Monte Carlo method by Silver et al. [58]. Like DQN, adopting DPG algorithms for process control task was a challenge. Due to their episodic-nature, Monte Carlo training methods are most suitable to handle finite tasks with explicit terminal states. In process control tasks, the implementation of DPG algorithm often lead to complex computation, which would potentially result in undesirable high variance. In an attempt to remove the limitations of deep RL for process control, the deep deterministic policy gradient (DDPG)

scheme that leverages the functionalities of DQN and DPG into one actor-critic RL algorithm was developed by Lillicrap et al. [59]. In DDPG the value function and policy are both approximated by DNNs [58]. Compared to previous method of deep RL, DDPG is naturally capable of addressing continuous control tasks successfully. Soon after the development of DDPG, Schulman et al. [60] proposed a deep RL algorithm known as trust region policy optimization (TRPO), which was intended for solving high-dimensional continuous control problems. TRPO remove the actor-critic framework of DDPG, identifying the control policies directly using update constraint. Conjugate gradient method was used for solving the optimal control problem in TRPO, subject to the policy constraint. However, data usage inefficiency and implementation difficulty of conjugate gradient method has raised concerns for real-time control applications of TRPO in practice [44]. To remove the limits of their original work, Schulman et al. [61], developed an improved version of the TRPO algorithm known as proximal policy optimization (PPO). In PPO update constraint was implemented directly, as a penalty, in the objective function using stochastic gradient descent method. Time line summary of the evolution of deep reinforcement learning schemes is shown in Figure 4.2 [44].

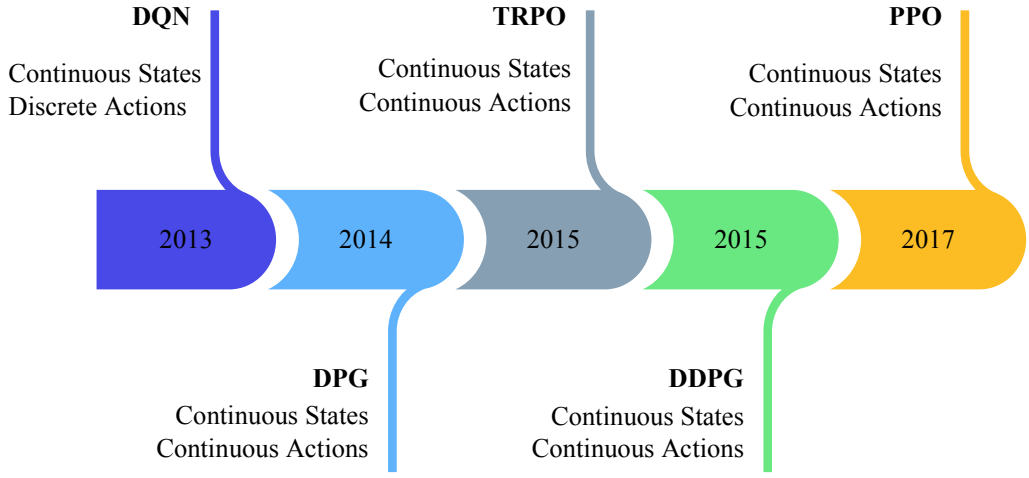


Figure 4.2. Timeline of deep reinforcement learning.

4.2. Deep Deterministic Policy Gradient (DDPG)

A policy-gradient (PG) algorithm is an end-to-end policy optimization procedure that computes noisy estimates of the gradient of the expected reward of the policy and then updates the policy along the gradient direction [62]. As such, PG uses a stochastic policy $\mu(a|s)$, which gives a

probability distribution over actions. DDPG is a variant of PG algorithm that combines a stochastic behavior policy and a deterministic target policy. It asses the policy, and then use the policy gradient to optimize performance. DDPG uses the actor-critic algorithm structure and is implemented with two DNNs to predict action and generate reward signal [62]. The network parameters' update rule for the critic and actor networks comes from the reward and deterministic policy gradient respectively.

In DDPG, the control policy μ is parametrized by a vector θ^{μ}. Thus learning of the optimal control policy means learning the optimal value for θ^{μ}. This is achieved by updating θ^{μ} in the direction that maximizes the performance function $J(\mu)$ as:

$$\nabla_{\theta^{\mu}} J(\mu) = \mathbb{E}[\nabla_a Q(\boldsymbol{s}, \boldsymbol{a}) \nabla_{\theta^{\mu}} \mu(\boldsymbol{s})] \tag{4.1}$$

As stated, the action value function Q and the control policy μ in DDPG are estimated by the critic and actor DNNs respectively.

In addition, target critic network $Q'(s, a)$ and target actor network $\mu'(s)$ having parameters $\theta^{\mu'}$ and $\theta^{Q'}$ are also implemented to stabilize the learning process [7]. These target networks are parameterized by the actor and critic networks such that they slowly track the network parameter changes as:

$$\theta^{Q'} = h\theta^{Q} + (1 - h)\theta^{Q'} \tag{4.2}$$

$$\theta^{\mu'} = h\theta^{\mu} + (1 - h)\theta^{\mu'} \tag{4.3}$$

Where the constant h is chosen to be a very small positive number.

The network parameters' update rule for the critic and actor networks is carried out using the error back propagation algorithm and deterministic policy gradient [7]. The governing equations for the target y_i and loss function L of the critic network are:

$$y_n = r_n + \gamma Q'(s_{n+1}, \mu'(s_{n+1}|\theta^{\mu'})|\theta^{Q'}) \tag{4.4}$$

$$L = \frac{1}{N}\sum_{n}(y_n - Q(s_n, a_n|\theta^{Q})^2) \tag{4.5}$$

Where $N, r \; and \; \gamma$ designate batch size of training data with n^{th} sample, immediate reward, and commutative reward. Therefore, the weights of the critic network are updated based on the loss function of Eq. (4.5). Figure4.3 illustrates the proposed DDPG based optimal control scheme

[7].The detailed DDPG learning algorithm implemented to handle the constrained optimization problem in this research work is summarized in Algorithm 4.1.

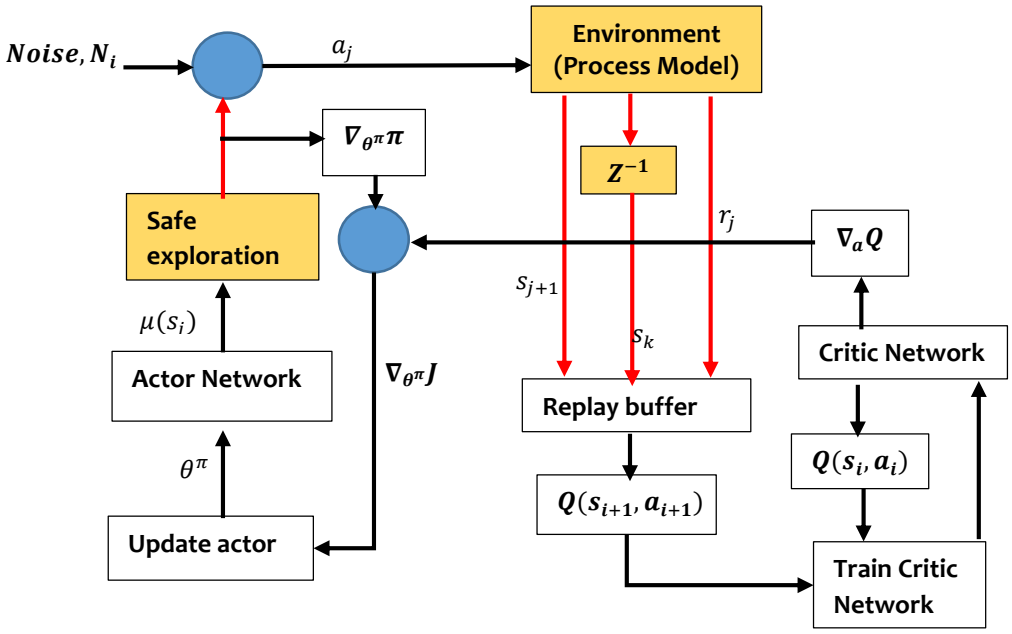


Figure 4.3. Schematics of the envisioned optimal control scheme adopted from [7].

Table 4.1. The steps in the actor-critic DDPG algorithm

1. Initialize the critic $Q(S,A)$ with random parameter values θ_Q, and initialize the target critic with the same random parameter values: $\theta Q' = \theta Q$.
2. Initialize the actor $\mu(S)$ with random parameter values θ_μ, and initialize the target actor with the same parameter values: $\theta\mu' = \theta\mu$.
3. For each training time step:
   - For the current observation S, select action $A = \mu(S) + N$, where N is stochastic noise from the noise model.
   - Execute action A. Observe the reward R and next observation S'.
   - Store the experience (S,A,R,S') in the experience buffer.
   - Sample a random mini-batch of M experiences (S_i,A_i,R_i,S'_i) from the experience buffer.
   - If S'_i is a terminal state, set the value function target y_i to R_i. Otherwise, set it to [Eq. (4.4)].
   - Update the critic parameters by minimizing the loss L across all sampled experiences [Eq. (4.55)].
   - Update the actor parameters using the sampled policy gradient to maximize the expected discounted reward [Eq. (4.1)].
   - Update the target actor and critic parameters depending on the target update method [Eq. (4.2)-(4.3)].

4.3. Benefits and Drawbacks of Deep RL for ISTSs

Some of the optimal control challenges of SIES that could be addressed using reinforcement learning approach are:

Centralized – decentralized optimal control: The RL approach allow realization of a general control structure, through modifying its reward function. In this sense, RL can be used to imitate a supervisory and/or regulatory controller(s) while including other measures such as economic in the reward function (Figure 4.4). This possibility could be used to remove the limitations of traditional hierarchical optimization scheme where local operation and system-level optimization operate at differing time scales [63]. Therefore, RL-based optimal control is suitable for the assessment of both centralized and decentralized strategies on the same ISTS.

Rapid online computation time: The offline pre-computation of RL's optimal policy allow the online evaluation extremely fast for implementation. Therefore it can be an effective and elegant solution for applications where offline computation time is not a problem [44].

Calibration as well as control: The use of RL agents as intelligent units in a general sense, allows the calibration of models and many types of controllers with different structures [44]. These models and/or controllers can then be incorporated in the same framework using standard interfaces. This, for example, allows calibration via iteratively adapting an empirical solar radiation model that could serve as a prediction (disturbance) model in optimal control.

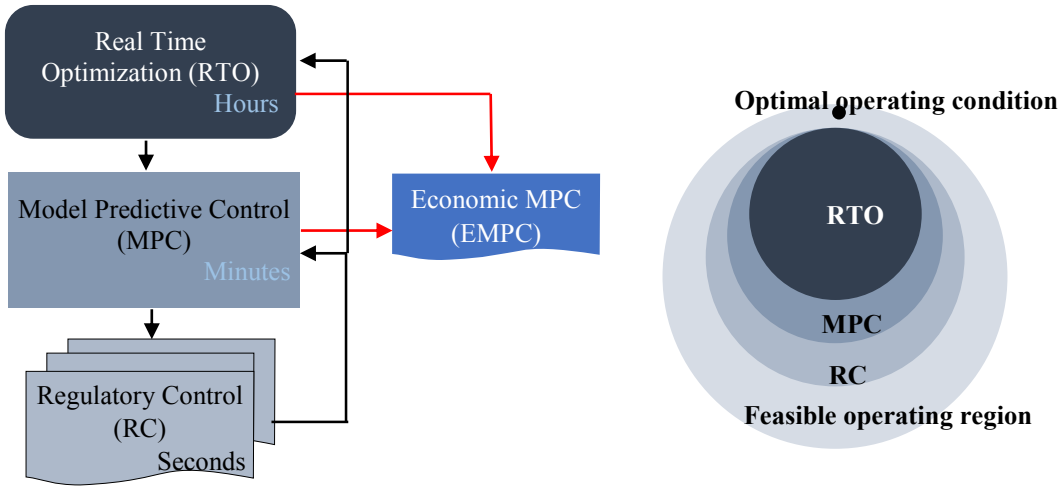


Figure 4.4. Prevailing optimal control in process industry showing hierarchical optimization (left), and range of control (right).

Adaptive model-free control: Model may only be necessary during training, which the RL agent discard during real-time implementation. Therefore can truly reproduce adaptive and evolutionary mechanisms [53]. Modelling based on agents focuses on individual units and therefore online model modification for offset free control, which might not work well for noisy processes, is not required, as it is the case in MPC. Through their reactivity to the environment, these models are dynamic in an unstable equilibrium. This feature of RL can used for optimal control where underling processes dynamics is complex for accurate modeling.

This aforementioned advantages could enrich the optimal control in ISTS by emulating controller (s) or enabling calibration including models for prediction such as solar radiation. However, this is not an easy task. Many challenges due to inherent drawbacks of RL methods arise, such as:

- How can we be sure that the performance of RL-based optimal control is better than conventional methods?
- How can we represent different control and/or calibration roles to the same RL algorithm in an adequate way?
- How do we know that training in RL, which usually take long time, eventually converges to the optimum control policy? What can we do to facilitate the training process?
- Is it worth the effort to train an RL agent, when there is significant constraint in terms of large and quality dataset requirement?

These issues will be addressed in the following section, through the simulation results of case studies. Some relevant findings are included to illustrate the advantages of using the selected approach. Inclusion of a prior knowledge about system dynamics, design of experiment for active dataset selection as well as efficient reward function formulations are discussed as remedy for RL drawbacks. Finally, insights drawn from these studies are presented, as well as a way for future research is indicated.

4.4. Case Study: Optimal Control of Distributed Solar Field

In this case study, the optimal control of Distributed Solar Field (DSF) is formulated as a constrained Markov decision process (CMDP), with the aim of maintaining collector outlet temperature within the acceptable range while minimizing thermal power losses. In this control problem, both the state and action spaces are continuous. A DDPG algorithm, not necessarily

knowing the DSF dynamics, is applied to learn the optimal control policies, which determine the optimal control actions from the states. To co-optimize temperature and energy goals, a mix of discrete and continuous reward signals are designed. These signals guide the learning agent to productive state-action space as well as improve convergence by providing a smooth reward near target states.

The CSP based on PTC in Daggett, CA, due to the availability of plant configurations and the performance data, is considered as the reference plant in this study [64]. Table C summerizes the relevant solar plant configurations of DSF. It should be noted that, this setting is not restrictive since the proposed DDPG control method can be directly extended to other solar plants with different configuration and concentrating technologies such as central receiver (Tower), linear Fresnel reflector (LFR) and parabolic dish (PD).

The Optimal Control Problem

The control objective is to maintain the desired temperature of the HTF while maximizing thermal energy collection under various disturbances. The co-optimization performance of the control scheme depend on its ability to learn and regulate the effects of main disturbances. The main disturbances in DSF are solar irradiance, inlet temperature of the HTF, and the ambient temperature (Figure4.5). The residence time of the HTF within the absorber pipe, which directly affects the heat accumulation rate from solar irradiation, varies in relation to flow rate. However the control strategy has to obey the maximum and minimum operational limit of the oil pump (Table B.1). Violation of these operational limits would potentially lead to undesirable result. For example, the lower operational limit reduce the risk of oil decomposition occurring at higher temperatures exceeding 400 °C, which may lead to oil change in the entire DSF [26]. In addition, the risk of oil leakage increases as inlet and outlet temperatures exceeds the normal value leading to high oil pressure in the pipe system [26].

Furthermore, the incidence angle has direct relation with the absorbed solar energy, and on the temperature of the HTF. At normal condition and perfect sun tracking the angle of incidence can be assumed to be zero to maximize energy collection. However, in real operation of the DSF, the orientation of the solar collector is manipulated to avoid over heating of the HTF [26].

To this end, the flow rate of the HTF and the position of SCAs are considered as the manipulated variables in DSF optimal control. In this work, it is assumed that the 4 SCA controllers work simultaneously in a similar manner (Figure4.5).

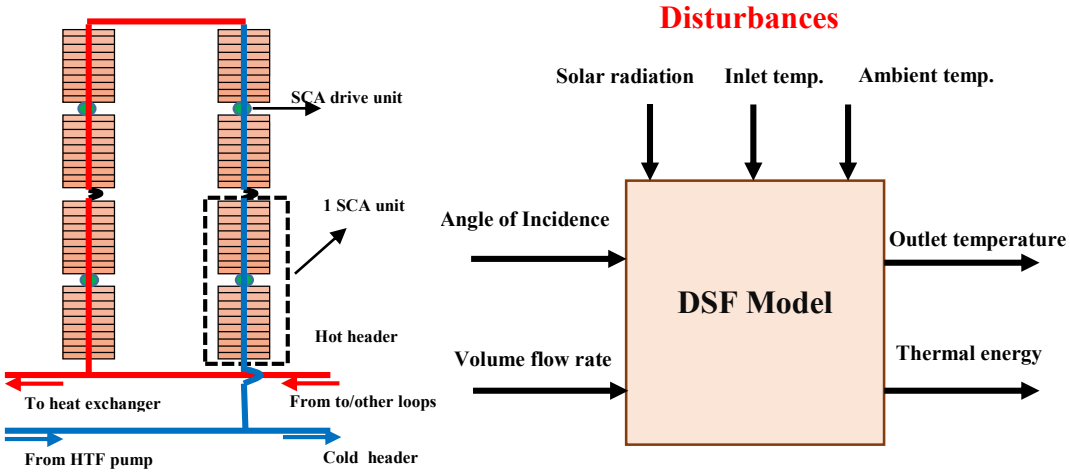


Figure 4.5. Schematics of SCA for DSF modelling.

Considering the aforementioned points, the DSF optimal control problem can be formulated as an optimization program, which is given by

$$\Omega = \min_{V_F,\theta} \Omega_1 + \Omega_2 + \Omega_3 \tag{4.6a}$$

Where

$$\Omega_1 = M \quad if\ (\varepsilon_T^2 + \varepsilon_V^2) < \alpha \tag{4.6b}$$

$$\Omega_2 = N \quad if\ (|\varepsilon_T| \geq \beta_1\ or\ |\varepsilon_V| \geq \beta_2) \tag{4.6c}$$

$$\Omega_3 = \delta_1(\varepsilon_T + \theta)^2 + \delta_2(\varepsilon_T - \theta)^2 + \delta_3\varepsilon_V^2 \tag{4.6d}$$

Subject to the constraints

$$V_{min} \leq V_F \leq V_{max} \tag{4.7a}$$

$$\theta_{min} \leq \theta \leq \theta_{max} \tag{4.7b}$$

In the above optimization problem formulation Ω_2 and Ω_2 are discrete cost function to drive the system away from bad states and keep it in the desired region. On the other hand, Ω_3 is a continuous cost function to improve convergence from the desired region to target operating points. The parameters α, β and δ are trade-off design variables to co-optimize temperature and energy goals. These goals are measured by the temperature tracking error, ε_T and flow rate deviation from the maximum value, ε_V.

Directly solving the above optimization problem is difficult, as it is not convex due to the peculiar characteristics of DSF. First, the DSF dynamics are not time invariant because of the time-dependent HTF flow rate. Second, the effects of multiple, simultaneous and inherently uncertain disturbances may not be easily determined in practice. To handle these difficulties, model-free deep reinforcement learning algorithm can be implemented by considering the optimization problem as a CMDP.

The DDPG algorithm

The architecture of the critic and actor networks are shown in Figure 4.6. The critic network consists of six hidden layers, which include four fully-connected layers and two ReLU layers. Its output is the reward. On the other hand, the actor network is composed of three hidden layers, which include two fully-connected layers, and a tanh layer. Its output is the control action.

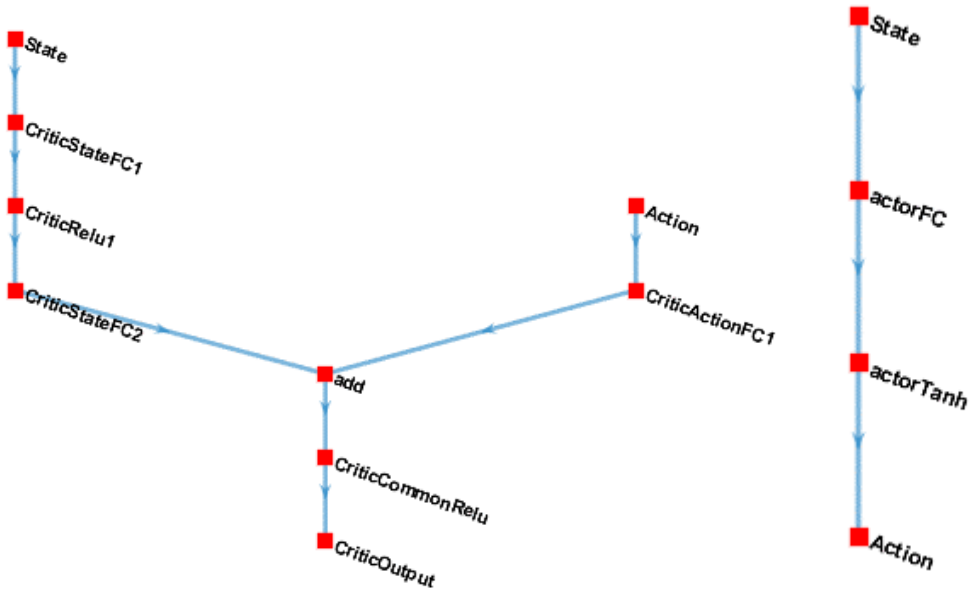


Figure 4.6. Architecture of the (left) critic network and the (right) actor network.

The detailed DDPG learning algorithm implemented to handle the constrained optimization problem is summarized in Table 4.1.

Simulation Experiment

For the simulation experiment, a DSF model and control methods are developed using existing Simulink library and components from the control and fuzzy logic toolbox. The Simulink model simulates three different controller subsystems, namely conventional MPC, PID-like nonlinear fuzzy logic controller (FLC) with triangular membership function [31] and the proposed DDPG

agent, to control the same DSF. However, a linear time-invariant model of DSF, which is used by the MPC, is initially developed following the indication given by Stautez [65] through a system identification experiment.

In an attempt to imitate the effect of real-time disturbances, direct normal irradiance (DNI) and ambient temperature data are taken from actual records from a metrological station [66]. Specifically the data are taken from the African Monsoon Multidisciplinary Analyses (AMMA) database. The chosen metrological station is BAMBA found at geographical position (17.099◦ N, 1.4018◦W and 280m altitude). The selected data for this work spans over the range of five years (2006–2010) measured at 15 minute interval. In order to generate smoothed disturbance values that suited the continuous control task, the discrete data have been interpolated with a monotone Hermite spline, between intervals of measurement. In addition, data consistency is inspected and any missing values are replaced by the average values in their vicinity (Figure4.7).

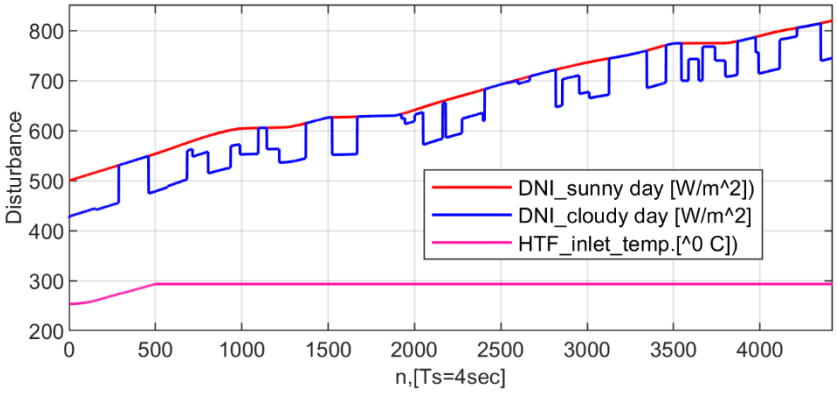


Figure4.7. Disturbances used in simulation experiment.

The simulation experiments are conducted in Matlab with Reinforcement Learning Toolbox. The hardware environment for the simulation uses an HP ProBook workstation with Intel Core i7-6500U CPUs, an AMD Radeon RX 540X graphic card, and 16GB RAM memory. Figure 4.8 depicts the schematics of the DDPG-based optimal control implementation in MATLAB/SIMULINK. The simulation experiment settings for training the DDPG agent is shown in Table 4.2. Insights form previous experimentation on deep learning reported in the literature are used for setting the initial DNN configurations. Afterwards these initial hyperparameters are adjusted through trial-and-error.

Table 4.2. Summary of the DDPG training settings.

Training options	Value
Smooth factor	0.001
Discount factor	0.95
Mini batch size	48/96
Maximum epochs(Typical days)	240
Critic network learning rate	0.001
Critic network regularization factor	1.0-04
Actor network learning rate	0.001
Actor network regularization factor	1.0-04

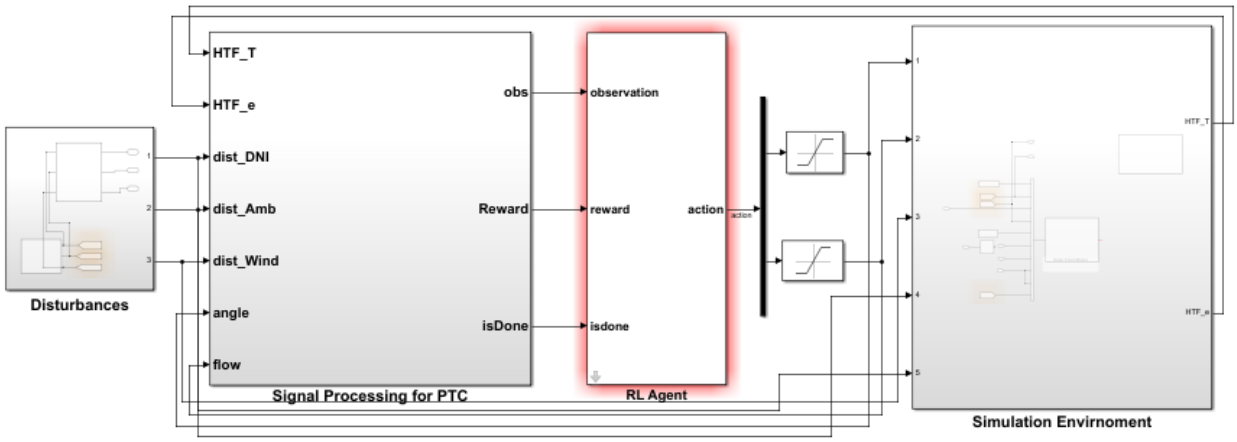


Figure 4.8. Schematics of the DDPG agent implementation in SIMULINK

For this model:

- The action signal consists of flow rate and incidence angle actions. The flow rate action signal takes value between 0.4 and 2.778 (m^3/s). The defocus action signal takes value between 0 degrees to 15 degrees.
- The reference temperature is set to be 666 K.
- The observations from the environment contain the temperature measurements: the temperature error,$e = T_{ref} - T_o$, its integral $\int e$ and model output temperature T_o. In addition, the observations contain the disturbance measurements: the ambient temperature deviation,e_1, inlet temperature deviation e_2, solar radiation deviation e_3 and their derivatives $\dot{e}_1$, $\dot{e}_2$and $\dot{e}_3$.
- The simulation is terminated when the HTF temperature exceeds the saturation limit or is less the inlet temperature.

- The reward r_k, provided at every time step k, is determined according to Eqn. 4.6 where $\alpha, \beta_1, \beta_1, \delta_1, \delta_2, \delta_3, M, N$ is set to the values 1,2,16,0.03,0.2,0.02,1,10 respectively.

The proposed DDPG algorithm is implemented to learn the control policy that adapt the incidence angle and HTF flow rate values so that it

- Result in adequate trainsient response and temperature set points tracking capability
- Balance the trade-off between reference tracking and flow rate (solar gain) maximization
- Address constraints for safe operation in DSF.

To this end, the DDPG agent is trained for a maximum of 1500 episodes. Referring back to reward signal definition, we see that it represent the set temperature deviation of the controller and the priority of controller action as well as the reward shaping penalty term. The trend of this reward value converging to optimal points is depicted in Figure 4.9. The relatively low reward value during the initial phase of training is attributed to fact that the DDPG agent starts with no a prior knowledge about the temperature regulation policy and tradeoff between the control actions. However, as the training continues, the DDPG agent becomes efficient at randomly selecting actions with a small noise leading to convergence of the reward value at about the 1195 th episode.

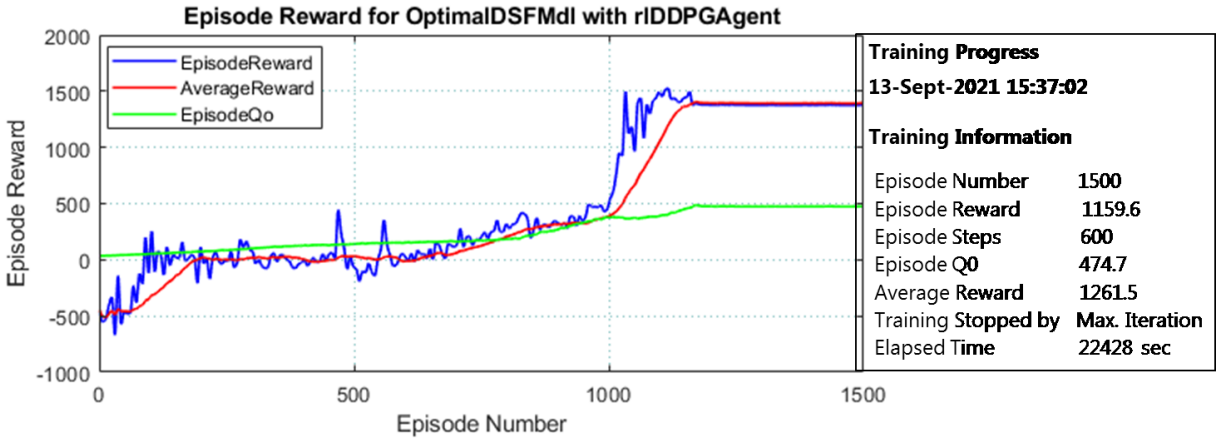


Figure 4.9. Training performance of the DDPG agent

After successful training of the DDPG agent, its control performances is analyzed by considering how the trained DDPG agent contrast to the above mentioned methods from the literature.

MPC vs DDPG

It should be underlined that one of the manipulated variables (HTF flow rate) relates directly to the net energy gain in DSF, but is a contradicting objective in the optimization problem to that of

temperature set point tracking (i.e. DSF response). To this end, the controllers' performance is measured based on their abilities at manipulating the control variables to co-optimize these contradicting objectives under the constraints of safe DSF operation.

Temperature tracking performance of the MPC and the proposed DDPG based control methods between 9 AM and 2 PM, for sunny day as well as synthetically generated cloudy day, is shown in Figure 4.10. In spite of the adequate temperature set point tracking without violating operational constraints, conventional MPC method has some drawbacks. First, the co-optimization capability of MPC under high penalty on violation of operational constraints is poor, as can be seen from the range of control actions. Secondly, when a significant change in the set point occurs, achieving good transient response is difficult as can be seen from the large rise time (leading to sluggish response).

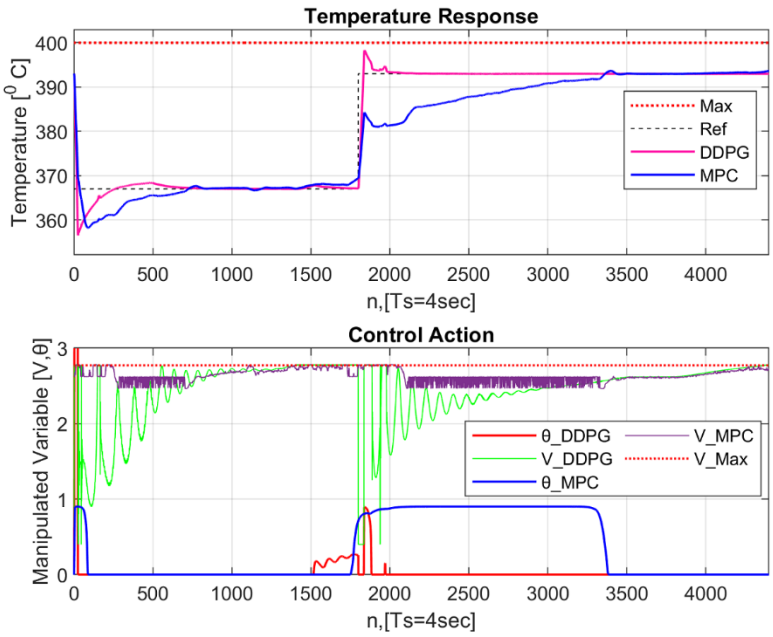


Figure 4.10. Temperature trajectories versus control actions of MPC and DDPG. DSF outlet temperature (top) and manipulated variables of HTF flow rate and SCA incidence angle (bottom).

Compared with the MPC controller, the control actions from the DDPG agent simultaneously improve both the temperature set points tracking and the flow rate deviations from the desired value of 2.77 m3/s. The responses trend for the change in set point are almost identical, which indicates that DDPG agent can control the DSF well despite changes in the system. This capability ensures that this control scheme is guaranteed to lead to optimum energy gain from the DSF. This capability is because the DDPG agent can create a nonlinear control surface and is

able to simultaneously manipulate flow rate and incidence angle more quickly when the error is large; and when the error is small, it becomes less aggressive to avoid violation of operational limit. Furthermore, as Figure 4.10 shows, the proposed DDPG agent is able to emulate an MPC controller and can also be used to enhance its performance. This similar and/or better control performance suggests that the advantage of the proposed approach lies in the lesser requirement for process model and less online computational effort. Consequently the DDPG-based constrained optimal control scheme outperforms the MPC method.

PID-like FLC vs DDPG

Similarily, Figure4.11 depicts the daily HTF flow rate and outlet temperature trajectory for the representative design day of May 1, 2008 for the PID-like FLC and DDPG based controllers. It is possible to see that the temperature response from the PID-like FLC oscillates between low and high values that occurs in two time periods, The first period occurs during early hours, when the DNI is low. The second period occurs near the solar noon, when the DNI is high. In these two periods, the competing effects of the HTF flow rate and orientation of SCAs result in high temperature oscillations that is level off after a while before settling to the refrence final value. As expected, the most problematic temperatures, that violate the maximum temperature limit, happens near the solar noon.

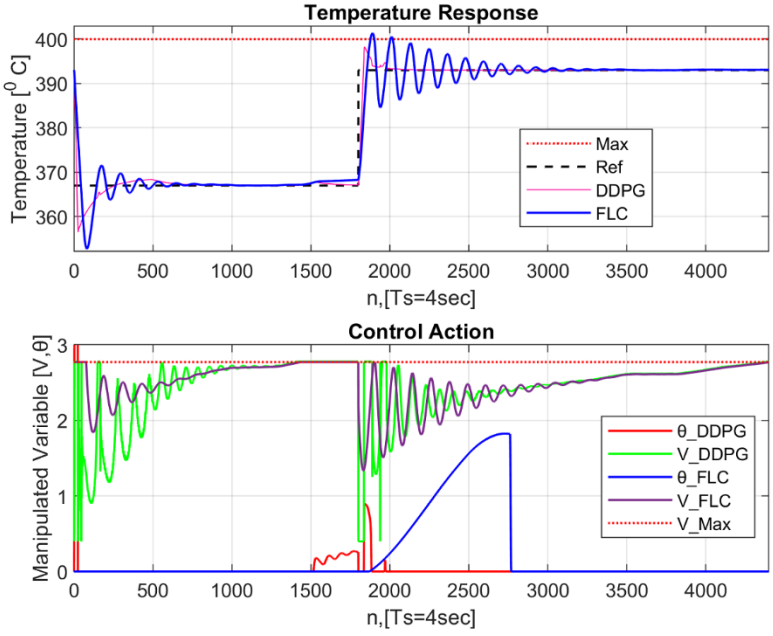


Figure 4.11. Temperature trajectories versus control actions of PID-like FLC and DDPG. DSF outlet temperature (top) and manipulated variables of HTF flow rate and SCA incidence angle (bottom).

Finally, the performance of the considered DSF control methods under active disturbance is compared. To this end, cloudy day temperature tracking performances is evaluated and depicted in Figure4.12. The poor control performances from both MPC and PID-like FLC is a direct consequence of the need for a adequate dynamic model and a bad estimation of the nonlinear DSF dynamics, under active disturbances, in these regions. This has also resulted in the same problematic control action from PID-like FLC, which violate the maximum temperature limit, happening near the solar noon. Notice also the similar and desirable response curve of the DDPG-based controller under passage of clouds. This similar control performance suggests that DDPG agent adapt as influence change. Therefore, the DDPG based constrained optimal control in DSF shows a superior performance over the whole operating region when compared to the remaining control methods.

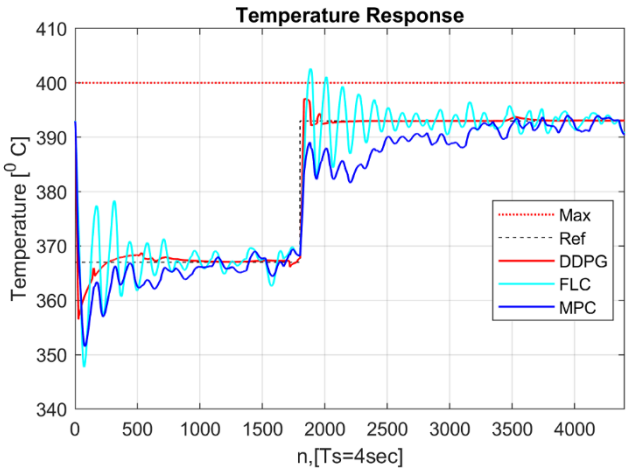


Figure 4.12. Cloudy day HTF outlet temperature profile.

Chapter 5

Solar Radiation Prediction via Empirical Model Calibration

Unlike conventional radiation plants, the primary energy source of a solar plant is both variable and uncontrollable, and acts as an active disturbance affecting its performance. Thus, interval forecasting of solar radiation that is suited to the requirements of robust optimization formulations and control schemes is vital. However, prevailing solar radiation prediction schemes often rely on simplifying assumptions and consequently have poor performance; or are too complex for integration to other models greatly limiting their practical use [67]. In this work, a computationally efficient framework for a direct and diffused solar radiation prediction is elaborated.

The approach iteratively adapt an empirical solar radiation model through deep reinforcement learning using only measured temperature values (Figure 5.1). The learning agent (the empirical solar radiation model that outputs diffused and direct solar radiation) interacts with the environment (context related to solar radiation such as ambient temperature). To this end, the model calibration process is formulated as a constrained error optimization problem, in which air temperature and error deviations of the empirical model are used as states and actions. For efficient problem solution, a DDPG algorithm is trained to learn the optimal calibration policies, which determine the optimum error deviation values from the states. In order to use the adapted model as a predictor, a day ahead temperature forecast based on deep long short-term memory (DLSTM) network is also developed. The results of the proposed framework significantly outperforms previous methods of solar radiation prediction. The novel model calibration scheme has more extensive application potential for real-time tuning of controllers.

The need for this kind of prediction scheme is related to the following benefits.

- **Cost:** Once the model is trained sufficiently, there is no need for solar radiation measurements except for performance evaluation form time-to-time reducing the data measurement and associated cost.

- **Flexible utilization:** Ambient temperature is adequate for learning the model calibration process. Therefore, if its measurement at desired resolutions are available, it is possible to zoom in/out in time (and forecasting solar radiation) for a given day. Furthermore, it is possible to calibrate the empirical model simultaneously both for diffused and direct solar radiation using global radiation values only.
- **Ease of integration to optimization models:** Consequently, the calibrated and simple empirical model can now be integrated to models performing other tasks such as optimization and control of solar plants with ease.

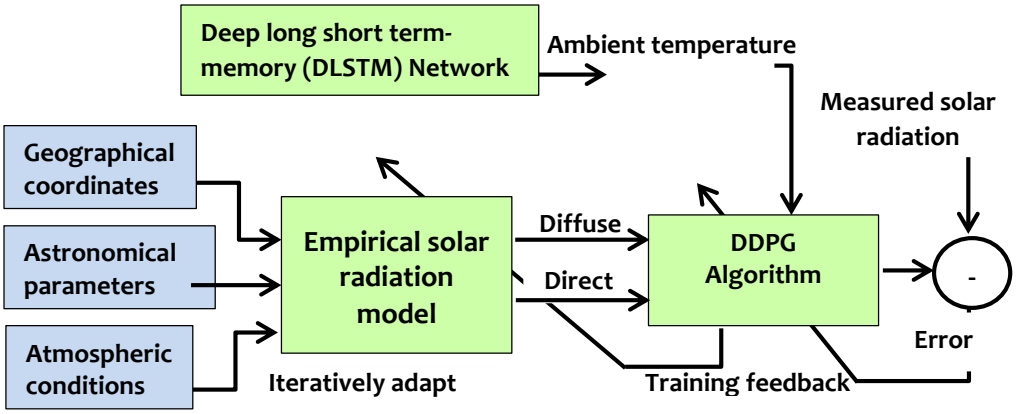


Figure 5.1. The schematic of the proposed virtual sensor

The relevant parts, as depicted in Figure 5.1, which enable prediction of direct and diffused radiation via empirical model calibration are described next.

5.1. Empirical Solar Radiation Model

The empirical solar radiation model estimates the hourly radiation, which would be received on a horizontal surface, under the "clear" sky condition. In reality, atmospheric conditions and air mass vary stochastically, thereby changing the estimated scattered and absorbed solar radiation. A model for approximating the beam radiation, τ_b, is based on zenith angle and altitude. The model also considers standard atmosphere conditions of different climate types. The beam radiation under clear sky condition can be formulated as [68]:

$$\tau_b = a_0 + a_1 \exp\left(\frac{-a_2}{cos(\theta_z)}\right) \tag{5.1}$$

The constantsa_0, a_1 and a_2 in the above expression are related to the altitude, Alt, at the given location and are determined from:

$$a_0^* = 0.4237 - 0.00821 * (6 - Alt)^2 \quad (5.2a)$$

$$a_1^* = 0.5055 - 0.00595 * (6.5 - Alt)^2 \quad (5.2b)$$

$$a_2^* = 0.2711 + 0.01858 * (2.5 - Alt)^2 \quad (5.2c)$$

To account for the influence of climate change, correction factors as given in Table 5.1 are also used. The clear-sky horizontal beam radiation is thus given as:

$$G_b = G_{on}\tau_b cos\theta_z \quad (5.3)$$

Where G_{on} for the i^{th} day is given by:

$$G_{0n} = G_{sc}\left(1 + 0.033cos\frac{360i}{365}\right) \quad (5.4)$$

Where the extraterrestrial radiation, G_{sc} is taken to be 1339 W/m^2

Table 5.1. Correction factors for climate types [68]

Climate Type	$\boldsymbol{r_0 = a_0/a_0^*}$	$\boldsymbol{r_1 = a_1/a_1^*}$	$\boldsymbol{r_2 = a_2/a_2^*}$
Tropical	0.95	0.98	1.02
Mid-latitude summer	0.97	0.99	1.02
Subarctic summer	0.99	0.99	1.01
Mid-latitude winter	1.03	1.01	1.00

In order to estimate the clear-sky diffuse radiation, G_d on a horizontal surface, the model developed by Liu and Jordan [67] is used and is given by:

$$G_d = G_o(0.271 - 0.294\tau_b) \quad (5.5)$$

Finally the global (total) clear-sky solar radiation, G_g is given by

$$G_g = G_d + G_b \quad (5.6)$$

5.2. Calibration and State Description

The proposed model calibration scheme uses historical measured values as reference and empirical model outputs, determined in Eq. 5.3 and Eq. 5.5, as adjustable model to mimic the actual diffused and direct solar radiation signals. In this work the model calibration process is

considered as a constrained error optimal control problem. At each time step, the actor generate actions influencing the environment, then the states of the environment such as air temperature change. The aim of the interacting actor is to find the optimal policy that results in maximum reward measured in respect to the extent of prediction deviation. Basically, the calibration process is a mapping of the model output deviation from measured values based on the states in the environment. The learning process which enable learning of the optimal calibration policy is driven by the reward signal given by:

$$\Omega = \min_{a_c^*} \Omega_1 + \Omega_2 + \Omega_3 \tag{5.7a}$$

Where

$$\Omega_1 = M \quad if\ (\varepsilon_b^2 + \varepsilon_d^2)\ <\ \alpha \tag{5.7b}$$

$$\Omega_2 = N \quad if\ (|\varepsilon_b| \geq \beta_1\ or\ |\varepsilon_d| \geq\ \beta_2) \tag{5.7c}$$

$$\Omega_3 = \delta E(a_c^* + \Delta a_c^*) + (1-\delta)\|a_c^*\|^2 \tag{5.7d}$$

Subject to the constraints

$$a_{min} \leq \Delta a_c^* \leq a_{max} \tag{5.8}$$

Similar to the DDPG agent implementation for optimal control in DSF, Ω_2 and Ω_2 are discrete cost function to drive the system away from bad states and keep it in the desired region. On the other hand, Ω_3 is a continuous cost function to improve convergence from the desired region to target operating points. The parameters α, β and δ are trade-off design variables to co-optimize the deviations of solar radaition for direct, ε_b and diffused, ε_V as well as global, E from meeasured values.

The author argues that only air temperature (measured or predicted) and empirical model output as independent input variables to the DDPG agent results in sufficiently accurate prediction. This is based from investigations on all relevant solar radiation prediction variables reported in the literature as well as validation through simulation experiment. In the literature, the relevant input variables that are identified include temporal variables of time and date as well as environmental conditions of relative humidity, air temperature and wind speed [69-70]. The temporal variables and spatial components are explicitly used in the empirical model that estimates the direct and defused solar radiation using Eq. 5.1-5.5, therefore, these variables are not directly taken as input

Air temperature, wind speed and relative humidity, based on the findings reported in the literature, are the remaining relevant solar radiation prediction variables with degree of relevance in that order. In fact the works of Hussain S. et al [69] state that wind speed and relative humidity have lower degree of relevance in solar radiation prediction (average contribution of about 15.2% and 24.8%) when compared to air temperature.

To demonstrate the adequacy of temperature as an independent input variable, different scenarios are constructed and analysed by considering how the type and number of relevant input parameters affect the prediction accuracy. In order to make fair comparisons, the maximum number of independent variables in the scenarios is limited to two (in addition to the empirical model output). For the comparative analysis a nonlinear autoregressive with exogenous input (NARX) neural network [71] with features summarized in Table 5.2 is used. Furthermore, training on the same NN model, which is reinitialized with the same random weights values generated while testing the first scenario are used. However, due to random initialization at the start of the experiment, in which the weight values significantly affect the network performance, replication of experiments are also carried out [72]. Thus, six replication experiments that initialize and train the NARX network are performed to select optimal weight initialization values. The training algorithm chosen is Levenberg-Marquardt (LM) as it provides a guaranteed performance minimization of the error function with relatively fast convergence rate [73].

Afterwards, five scenarios are defined. The empirical model is designated as a reference scenario (M0) and is also used in the remaining four scenarios. The additional scenarios are set as follows. Temperature being highly correlated to solar radiation represent one scenario while also being used jointly with wind speed, humidity, and their combination to construct the remaining scenarios. A scenario for relative humidity along with temperature and wind speed is not consider in this work for two reasons. First, although there is a prediction performance improvement, if this scenario is incorporated, it is not significant [69-70]. Second due to their inherent stochastic nature, the accuracy of the solar radiation prediction drastically decline as the number of forecasted weather condition it depends on increases. Therefore, the prediction scheme assumes that a maximum of two independent input variables other than the empirical model should be sufficient. Table 5.3 gives summaries of these scenarios.

It is important to note that the empirical model that is present in all the scenarios is always initially known. In addition, the same data set that is divided in to training (70%), testing (15%) and validation (15%) is used in the comparative study. The dataset used for each scenario training (and how it is collected) is the same as that for the implemented DDPG agent, and is discussed in section 5.4 of the experimental simulation setup

Table 5.2. NARX network architecture for scenario comparisons

NN Type	Architecture	#Input /Output Feedback	Training Algorithm
NARX	[30 10]	[1:8;1:8]	Levenberg-Marquardt

The following statistical metrics are used to evaluate the performances of the scenarios.

$$MABE = \frac{1}{N}\sum_{n=1}^{N}\left|G_{b,d} - \hat{G}_{b,d}\right| \tag{5.9a}$$

$$RMSE = \sqrt{\sum_{n=1}^{N}\frac{(G_{b,d}-\hat{G}_{b,d})^2}{N}} \tag{5.9b}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N}\left(G_{b,d} - \hat{G}_{b,d}\right)^2}{\sum_{n=1}^{N}\left(G_{b,d}\right)^2} \tag{5.9c}$$

$$\bar{R}^2 = R^2 - (1 - R^2)\frac{r}{N_{tt} - r - 1} \tag{5.9d}$$

where mean absolute bias error is MABE, root mean squared error is RMSE, coefficient of determination is $R^2$and adjusted coefficient of determination is $\bar{R}^2$. N_{tt} is the total number of data points in test data set, r is the total number of regressors in the network. Coefficient of determination indicates the variation of neural network response against the observed values. The larger the R-squared is, the more variability is explained by the linear regression model. The adjusted coefficient of determination is more useful for comparing models with a different number of predictors.

Table 5.3. Scenario definitions

Scenario	Scenario label	Model input variables
0	Mo	Time, date, astronomical and geographical parameters
1	MHT	EMO, relative humidity, temperature
2	MT	EMO, temperature
3	MHW	EMO, relative humidity, wind speed
4	MTW	EMO, temperature, wind speed

The generalization capability of the various scenarios based on evaluation on an entirely different test data is depicted on Figure 5.2. The depiction in Figure 5.2 (left) shows the statistical metrics of testing (MABE, RMSE, and $\bar{R}^2$) for the different scenarios. Positions of the circles on xy plane indicate the error values, MABE and RMSE. Colours of the circles indicate the values of $\bar{R}^2$. Taking this into considerations, scenarios constructed with temperature perform very well.

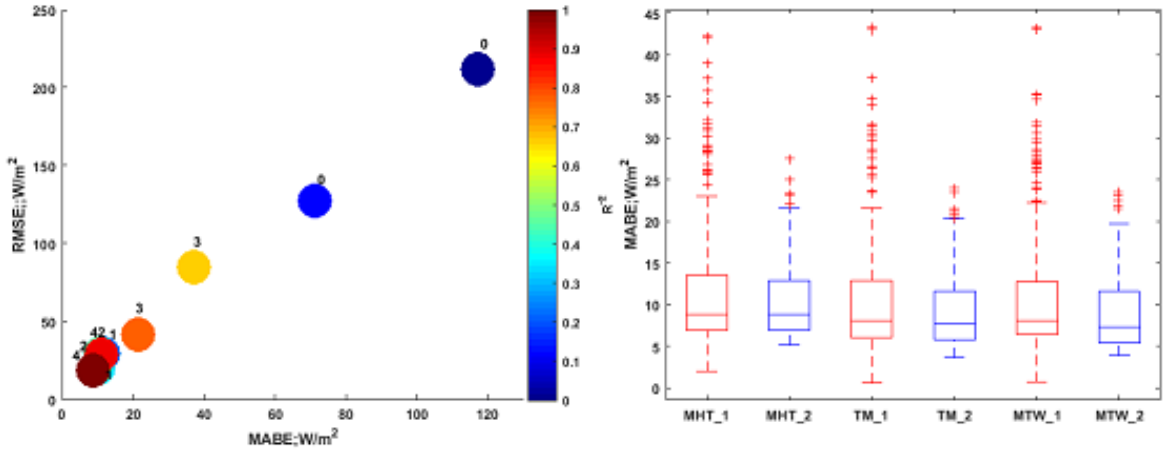


Figure 5.2. Comparative analysis of all (left) and of high performing scenarios (right).

On the other hand, the depiction in Figure 5.2 (right) indicates the performances of the selected highly performing scenarios, using a plot of prediction deviation, aggregated daily during the entire test year. From the above observations, it is possible to draw important conclusions. Scenario 2 with only temperature as an independent variable has similar performance as of scenario 4 and even surpass scenario 1 in terms of average and range of prediction error deviation. In addition, considering MABE outliers as indicated by the error observations beyond the whiskers, scenario 2 performs better than scenario 1. Furthermore, accuracy improves as the prediction method rely on the forecasting of fewer variables. Moreover, wind speed has higher

stochastic nature than ambient temperature, which makes its forecasting error higher. Relying on these findings it is clear that scenario 2 represent the best learning state for the DDPG agent.

Therefore, five features as factors of solar radiation prediction, are identified for training the DDPG agent at each time step.

- T, temperature between consecutive time steps
- ΔT, signed temperature difference between consecutive time steps
- $cos(\theta_z)$, Zenith angle
- Mean empirical model output error

5.3. Day Ahead Temperature Forecast

In order to use (train) the DDPG agent as a forecasting tool, it is necessary to predict on future values of the state i.e. air temperature at a given location. To handle this, a deep long short term-memory (DLSTM) neural network is trained. For this application, a sequence-to-sequence regression DLSTM network is used (refer Appendix C). The model inputs are observed temperature values and the targets are the training sequence with values shifted by one time steps. In the training process of the DLSTM, an adaptive moment estimation algorithm [74] with a piecewise learning schedule is used to minimize the root mean square error (RMSE) of the model output and actual historical values. Once the training process of the DLSM network is over, MATLAB built in function 'predictAndUpdateState' is used for the day ahead temperature forecast while updating the network state at each prediction. State update of DLSTM model via measurement is vital to enhance accuracy of temperature forecast, and of the DDPG agent.

5.4. Simulation Experiment, Results and Discussion

To assess the effectiveness of the proposed predictive deep reinforcement learning scheme, experimental simulations are carried out in this section as follows. First, the simulation setup is explained. Afterwards, the simulation results of the training performance of the solar radiation model and its prediction efficiency coupled to the day ahead temperature forecast are presented and analyzed. The scalability of the proposed scheme is also investigated for a refined time step of prediction. Finally, the performance of the solar predictor is compared with other schemes reported in the literature.

Experimental simulation setup

Metrological data from the African Monsoon Multidisciplinary Analyses (AMMA) database [145] are used for the simulation. The selected metrological station is BAMBA found at geographical position (17.099◦ N, 1.4018◦W and 280m altitude). The selected data for this work spans over the range of five years (2006–2010) and contains air temperature (T:◦C), relative humidity (RH: %), wind speed (WS: m/s), global solar radiation (G_g: W/m2) and diffused solar radiation (G_d: W/m2) measured at 15 minute interval. Data consistency is inspected and any missing values are replaced by the average values in their vicinity. The four year data are used for training while random samples are taken from the final year data for performing independent test of the trained DDPG and DLSTM.

The configuration and hyperparameter values for the DDPG agent is the same as that used for the optimal control case study. On the other hand, the hyperparameters of the DLSTM model is shown in Table 5.4.

Table 5.4. Hyperparameter values of the DLSTM model.

Training options	Value
Gradient Decay Factor	0.9
Initial Learn Rate	0.005
Learn Rate Schedule method	'piecewise'
Learn Rate Schedule drop rate factor	0.2
Learn Rate Schedule drop period	125
L2 Regularization	1.0-04
Gradient Threshold Method	'l2norm'
Gradient Threshold	1
Maximum Epochs	250
Minimum Batch Size	128

Temperature Forecast

Figure5.3 shows the RMSE performance of the DLSTM network trained on a 4 –year, 15-minnute interval temperature data based the setting summarized in Table 5.4. As can be seen from Figure5.3, the error overshoots at the start of the training and subsequently receded to a stable lower value. The LSTM model training process eventually lead to a low steady error starting at about and beyond the 210 iteration. Once the DLSTM network is trained, we forecast the air

temperature values for multiple time steps in the future. In order to do this, the state and update prediction function is run in a loop for the desired number of time steps one at a time, while updating the network state at each prediction. This way, the previous prediction is used as input to the function in order to get the next time step value and loop over for the remaining prediction. The result of this procedure for the day a head temperature forecast together with the associated errors at each time step is also depicted on Figure5.4. As can be seen from this figure, the forecast resembles the actual trend of air temperature throughout the day and the overall forecasting error, which is lower than 0.9, is also very low.

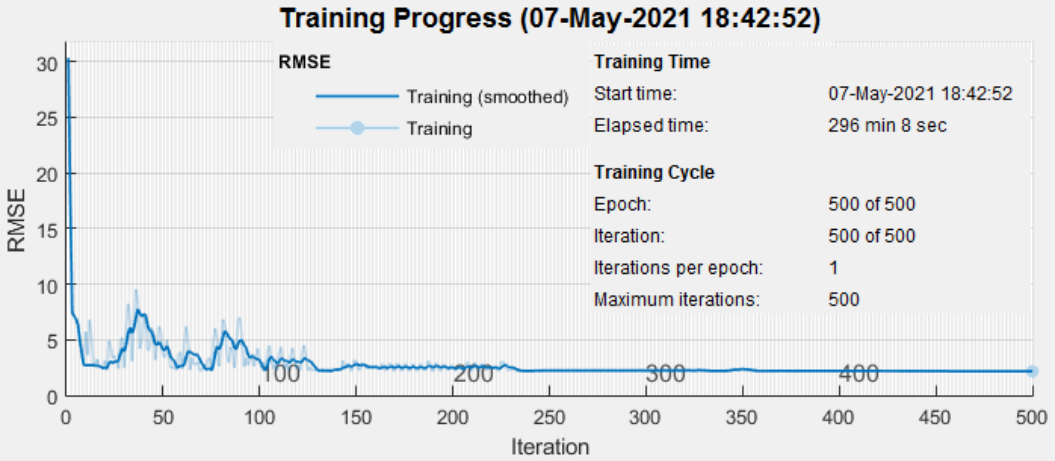


Figure 5.3. Training error (RMSE) of the DLSTM.

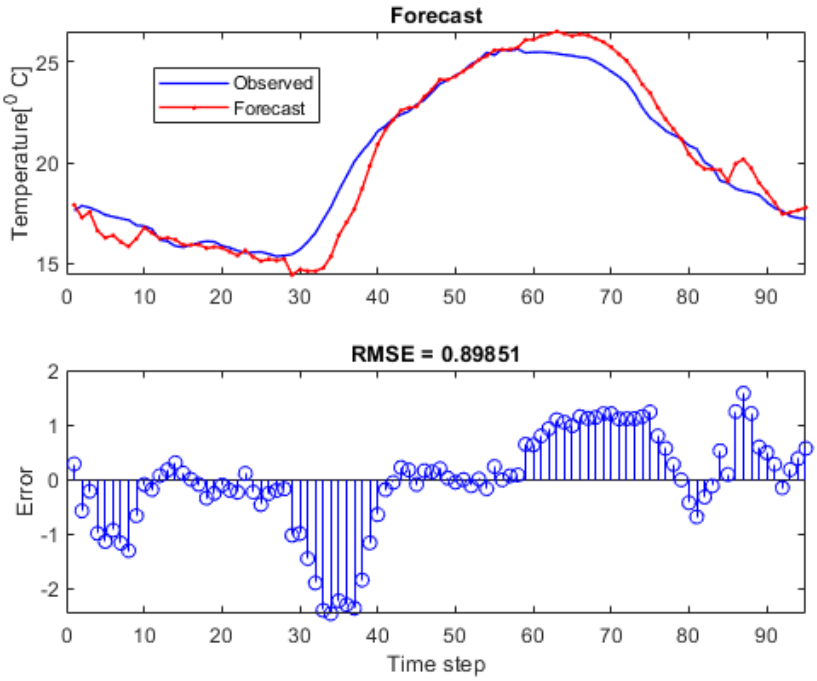


Figure 5.4. Day ahead temperature forecast performance.

Performance of the Calibrated Model

Carrying out the constrained error optimization problem encompassing all solar radiation data points at an acceptable time scale is practically infeasible. Thus, to facilitate the training of the DDPG agent, active datasets are selected based on Algorithm 2.1. Afterward, the proposed DDPG algorithm is implemented to learn the calibration control policy that adapt the empirical solar radiation model outputs closer to the observed values. The DDPG agent is trained with 1500 epochs using the algorithm explained in Section 4.2. Referring back to definition in (12), we see that there are three logical terms in the reward signal. These terms encourage the agent to make the mean absolute errors for the diffused, direct and global solar radiation small and ensures the stability of the calibration process while discouraging large deviations. The trend of this reward value converging to optimal value at about the 1470 th episode is depicted in Figure 5.5.

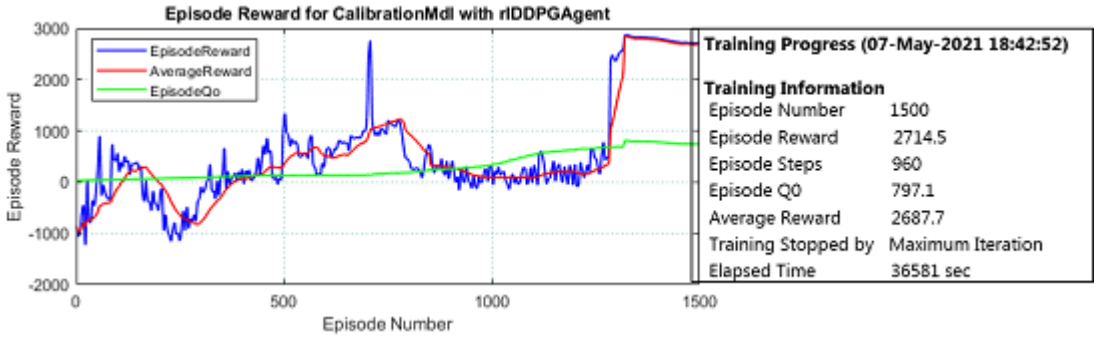


Figure 5.5. Evolution of the DDPG training for the model calibration.

After successful training of the DDPG agent, its prediction performances tied to the day ahead temperature forecast is considered. The aim of this simulation experiment is to evaluate the practicality of the proposed framework by considering how temperature forecast and its update impact the solar prediction accuracy. Figure5.6 shows the performance of the trained agent for, a 15-minute interval, direct solar radiation prediction of January 1, 2010. Obviously, the solar radiations that are further away from the solar noon are less fluctuating resulting in a relatively higher performance of the DDPG agent for these regions. Thus Figure5.6 show a relatively low error of prediction by the trained DDPG agent for these regions. Furthermore, Figure 5.6 also indicates the possibility for lesser overall error of prediction (about 5.2%) with temperature measurement updating the DLSTM model. Similarly, Figure5.7 show the performance of the DDPG agent for diffused solar radiation prediction. Overall, it can be seen that the agent learns well calibrating the empirical solar radiation model outputs closer to the observed values.

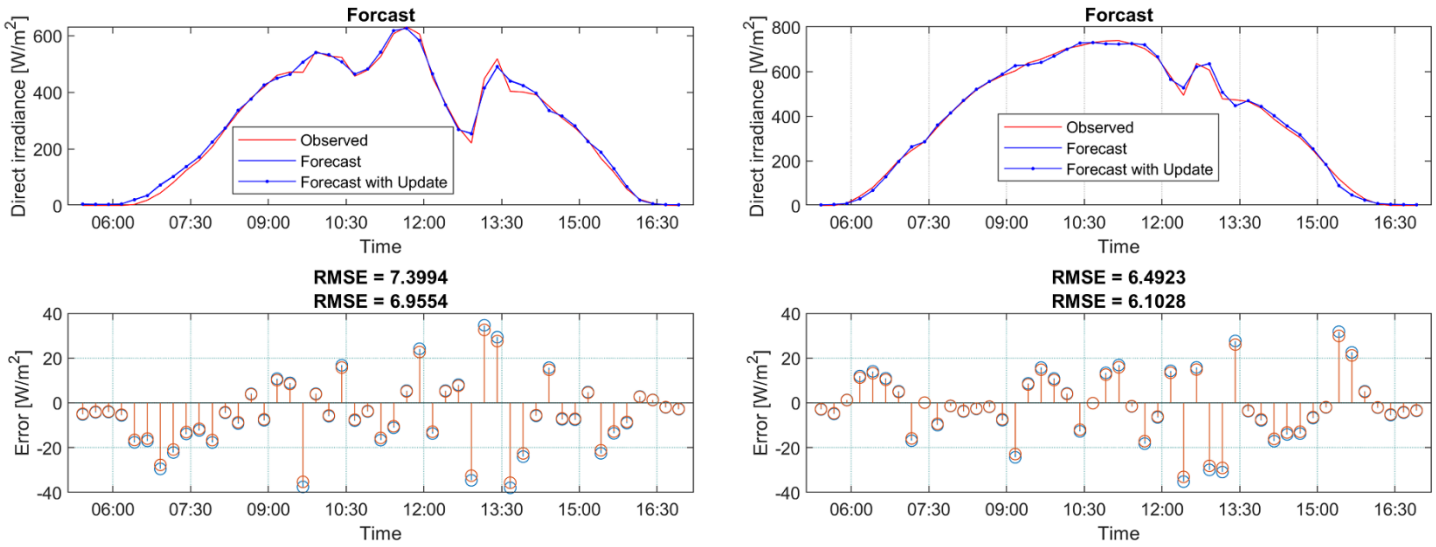


Figure 5.6. Sunny (left) and cloudy day (right) day ahead direct solar radiation prediction with and without temperature update.

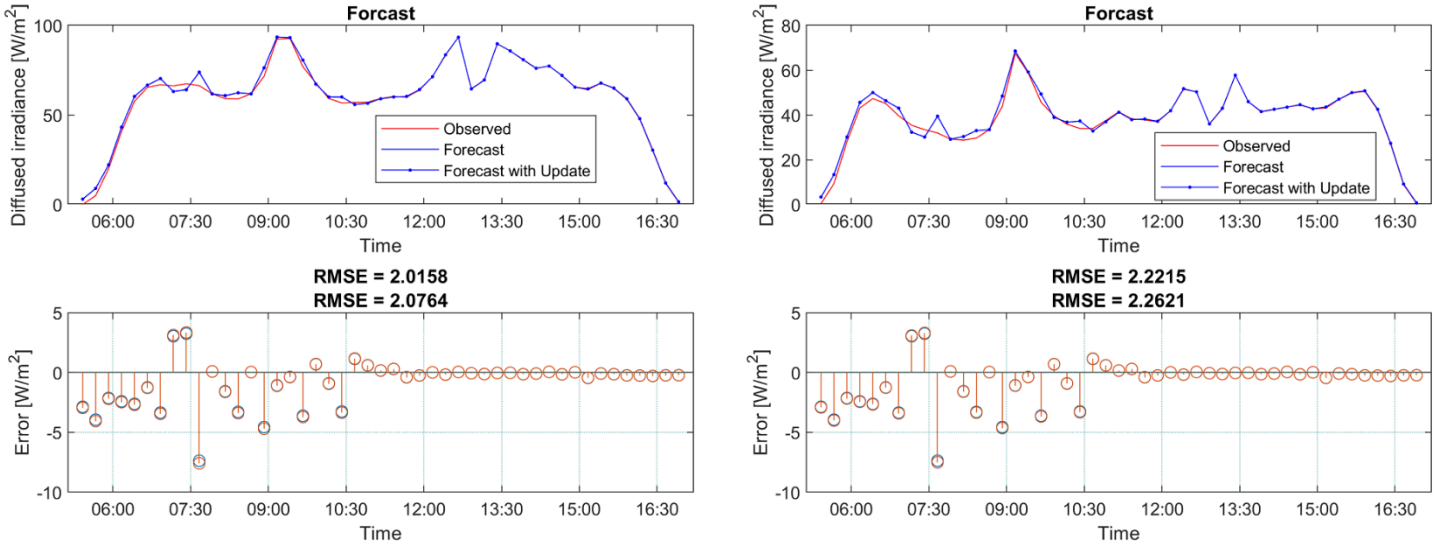


Figure 5.7. Sunny (left) and cloudy day (right) day ahead diffused solar radiation prediction with and without temperature update.

Performance Comparison

Here, the effectiveness of the proposed frame work is analysed by considering how the training prediction efficiency contrast to other methods of solar radiation prediction. Due to the absence of both direct and diffused solar radiation prediction from previous works and different intervals of predictions as well as metrological conditions, it is not possible to directly compare results from the previous works with our proposed scheme. To handle this limitation and also demonstrate the high prediction performance of our method, similar model configurations as of the aforementioned studies are implemented and trained on the same metrological data.

To this end, a DNN with two LSTM layers and two fully connected layers having 125 and 100 processing units, designated as DLSTM_S [75], is considered for prediction performance comparison. The output layer of the DLSTM_S was the regression layer, which outputs the solar radiation estimate. The training settings for the DLSTM_S is similar to the temperature forecast model used in this study. An ARIMA model, based on previous works in [76] is also developed to forecast solar radiation. Finally, a NARX with 30 hidden neurons and tan-sigmoidal activation function trained by Levenberg-Marquardt algorithm is designed. The NARX model is trained on the combined outputs of the ARIMA and DLSTM_S networks. Table 5.5 presents the RMSE performance for the training and day ahead forecasts of the proposed DDPG-based scheme and considered methods.

The RMSEs obtained using the ARIMA model is smaller than the DLSTM_S network. On the other hand, the NARX trained on the combined outputs of the DLSTM_S and ARIMA models has improved the day ahead prediction efficiencies of the separate models by over 50% for both the diffused and direct solar radiation. Judged by the error performance of Table 5.5, it can be inferred that the proposed DDPG based framework has superior direct and diffused solar radiation prediction efficiency for both training and test datasets.

Table 5.5. Prediction efficiency result of the comparative study

	Training error, RMSE (W/m^2)		**Test error, RMSE (W/m^2)**	
Prediction scheme	**Direct**	**Diffused**	**Direct**	**Diffused**
LSTM_S	189.20	101.62	211.52	127.54
ARIMA	217.22	121.34	256.01	177.13
NARX	84.943	41.72	106.59	71.10
DDPG	10.65	6.78	11.03	8.15

Chapter 6

Conclusions

The aim of this dissertation was to present the role and potential of a machine learning method to the design and control optimization of STSs in process industry. This was first considered from an algorithm and model perspective, explaining the basics of machine learning, which is rarely applied to industry-integrated STSs. Machine learning as an interdisciplinary field has been applied to optimize the design and performance of energy systems in hybrid vehicles, smart grids, buildings and other systems. Its capability of addressing the challenges of STS to industry integration is also evident as similar dynamic interactions and uncertainties can be found. Additionally, the STS system is heterogeneous and is coupled to other industrial integration aspects such as demand management and flexibility of utilization, which requires a holistic design and control optimization approach. This section concludes the findings and discuss their implications for addressing the STS-to-process industry integration challenges. Additionally, possible future research directions complementing the work are pointed out.

6.2. Summary of Findings

During this research work, a modular optimization scheme in ISTS was proposed along with evaluation of different performance indicators. First a ML based multi-modelling approach is used to create heterogeneous optimization interfaces, through combining stochastic, generic and empirical models. This scheme allows for disaggregated modelling of several similar technologies (and processes) and parameterize their inputs and local condition differently. Using this method, it is also possible to represent DER's and their additional capabilities of interactions. These are essential features that cannot be offered by conventional methods, which can be used to improve synergy and unlock the potentials of DERs in ISTSs. Particularly in this work, these interfaces led to the achievement of important findings from the perspective of design and control optimization.

In the dying process case study, it has been shown that layers in the optimization allow design improvements. Selection of few representative days, for random sets of optimum generation profiles, enables operation-based optimization at refined time scale. This has resulted in three capabilities not previously attained by conventional methods. First, it solves the design problem without compromising the objectives, and thus removes the limitation which makes computation of the optimal solution intractable. Second, it results in a holistic framework to simultaneously address both TES integration and its control concept optimization (for the efficient operation of STSs in process industry). Finally, it demonstrate the active role of demand, in which a process can have effects at a system level (solar gain) and influence energy efficiency. This indicate that integration of effective DSM policy in the design optimization should be explored further.

On the other hand, the optimal control case study in DSF demonstrate the capability of the ML approach to address performance limitations of conventional controllers. Set point tracking, co-optimization of objectives and ensuring safe operation are used as key performance indicators. The absence of adequate model in MPC (due to the linear time-invariant model) result in suboptimal solutions in terms of temperature tracking and net energy gain in DSF. On the other hand, the absence of robust disturbance prediction in the PID-like FLC leads to large temperature oscillations, which violate the maximum operational limit in DSF. In comparison, the DDPG agent was capable of learning disturbances while solving robust optimization formulations for constrained optimal control task. This finding demonstrate that ML can be leveraged for predicting both system dynamics and uncertainties, and can augment prevailing model-based control methods. Therefore, this dimension needs to be investigated further.

The other relevant finding was discovered from the solar radiation prediction case study (under sparse measurements coming only from ambient temperature sensor). In the case study, an empirical model was first used for joint metadata modelling (including feature selection) and training of a DDPG agent. Then the model itself was iteratively calibrated to serve as an efficient forecasting tool. It should be underlined that this forecasting tool is able to address some of the main challenges of STS integration to industry. This is especially so because of its reduced implementation and computation costs. However, the relevance of the scheme as a method of correction, for predictive control or energy management, was not considered, and need research attention.

6.2. Further research

In addition to the suggestions stated in the findings, other research possibilities are also anticipated. Comparative studies with other STSs in different process industries using similar approach could allow further validation of the work and identification of additional findings. In addition, developing a totally integrated optimization framework through defining the structures of the ML based model interfaces should be explored further. This would remove concerns regarding the applicability of the approach in practice. This is especially true because there is more data coming from the growing number of monitored STSs in process industries. Access to reliable data may provide solutions to address barriers and improve energy cost competitiveness of STSs. ML can be applied to distill information and capture uncertainties, which allow descriptive and diagnostic analytics. Should the measurements coming from different sensors lead to big data – in the sense of volume, velocity and variety of data– or there is multi-objective requirement, computation within reasonable time horizons is possible. Therefore, ML can be used to reliably design and manage an increasingly complex STS, and solving its challenges for industrial applications.

References

[1] Ravi Kumar, K., Krishna Chaitanya, N. V. V., & Sendhil Kumar, N. (2020). Solar Thermal Energy Technologies and its Applications for Process Heating and Power Generation – A Review. Journal of Cleaner Production, 125296.

[2] Wallerand AS, Kermani M, Voillat R, Kantor I, Mar_echal F. Optimal design of solar-assisted industrial processes considering heat pumping: case study of a dairy. Renew Energy 2017.

[3] Bergsteinsson, H. G., Møller, J. K., Nystrup, P., Pálsson, Ó. P., Guericke, D., & Madsen, H. (2021). Heat load forecasting using adaptive temporal hierarchies. Applied Energy, 292, 116872.

[4] Tilahun, F. B., Bhandari, R., & Mamo, M. (2019). Design optimization and control approach for a solar-augmented industrial heating. Energy.

[5] Julian D. Osorio, Zhicheng Wang, George Karniadakis, Shengze Cai, Chrys Chryssostomidis, Mayank Panwar, Rob Hovsapian, Forecasting solar-thermal systems performance under transient operation using a data-driven machine learning approach based on the deep operator network architecture, Energy Conversion and Management, Volume 252, 2022,115063.

[6] Stefan Hess. Low-Concentrating, Stationary Solar Thermal. Collectors for Process Heat Generation, PhD Dissertation, 2016.

[7] Kou, P., Liang, D., Wang, C., Wu, Z., & Gao, L. (2020). Safe deep reinforcement learning-based constrained optimal control scheme for active distribution networks. Applied Energy, 264, 114772. doi:10.1016/j.apenergy.2020.114772.

[8] Vzquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: a review of algorithms and modeling techniques. Appl Energy 2019;235:1072–89.

[9] Yan Z, Xu Y. Data-driven load frequency control for stochastic radiation systems: a deep reinforcement learning method with continuous action search. IEEE Trans Radiation Syst 2018;34(2):1653–6.

[10] Mocanu E, Mocanu DC, Nguyen PH, Liotta A, Webber ME, Gibescu M, et al. On-line building energy optimization using deep reinforcement learning. IEEE Trans Smart Grid 2019;10(4):3698–708.

[11] ESMAP. 2020. Global Photovoltaic Power Potential by Country. Washington, DC: World Bank. https://globalsolaratlas.info/global-pv-potential-study

[12] Baniassadi, A., Momen, M., Amidpour, M., & Pourali, O. (2018). Modeling and design of solar heat integration in process industries with heat storage. Journal of Cleaner Production, 170, 522–534.

[13] Ravi Kumar, K., Krishna Chaitanya, N. V. V., & Sendhil Kumar, N. (2020). Solar Thermal Energy Technologies and its Applications for Process Heating and Power Generation – A Review. Journal of Cleaner Production, 125296.

[14] Cotrado, M., Dalibard, A., Söll, R., & Pietruschka, D. (2014). Design, Control and First Monitoring Data of a Large Scale Solar Plant at the Meat Factory Berger, Austria. Energy Procedia, 48, 1144–1151.

[15] Crespo, A., Barreneche, C., Ibarra, M., & Platzer, W. (2018). Latent thermal energy storage for solar process heat applications at medium-high temperatures – A review. Solar Energy.

[16] Tobias Eiholzera, Donald Olsena, Sebastian Hoffmanna, Barbara Sturmb, Beat Wellig. Integration of a solar thermal system in a medium-sized brewery using pinch analysis: methodology and case study. Applied Thermal Engineering, 2017; 113:1558-1568

[17] Chaimaa El Mkadmi and Arifeen Wahed. Optimization of a solar thermal system for low temperature industrial heating process. IEEE Renewable and Sustainable Energy Conference (IRSEC), 2016: 14-17.

[18] Bet Sarkis, R., & Zare, V. (2018). Proposal and analysis of two novel integrated configurations for hybrid solar-biomass power generation systems: Thermodynamic and economic evaluation. Energy Conversion and Management, 160, 411–425.

[19] Kumar, L., Hasanuzzaman, M., & Rahim, N. A. (2019). Global advancement of solar thermal energy technologies for industrial process heat and its future prospects: A review. Energy Conversion and Management, 195, 885–908.

[20] Ghimire, S., Deo, R. C., Raj, N., & Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. Applied Energy, 2019; 253, 113541.

[21] Bahlawan, H., Morini, M., Pinelli, M., Poganietz, W.-R., Spina, P. R., & Venturini, M. (2019). Optimization of a hybrid energy plant by integrating the cumulative energy demand. Applied Energy, 253, 113484.

[22] Habibollahzade, A., Houshfar, E., Ahmadi, P., Behzadi, A., & Gholamian, E. (2018). Exergoeconomic assessment and multi-objective optimization of a solar chimney integrated with waste-to-energy. Solar Energy, 176, 30–41.

[23] Ghasemi, A., Heidarnejad, P., & Noorpoor, A. (2018). A novel solar-biomass based multi-generation energy system including water desalination and liquefaction of natural gas system: Thermodynamic and thermoeconomic optimization. Journal of Cleaner Production, 196, 424–437. doi:10.1016/j.jclepro.2018.05.160

[24] Krummenacher, P., Muster-Slawitsch, B. Methodologies and software tools for integrating solar heat into industrial processes. 13th International Conference on Sustainable Energy technologies, 2014.

[25] Sandá, A., Moya, S. L., & Valenzuela, L. (2019). Modelling and simulation tools for direct steam generation in parabolic-trough solar collectors: A review. Renewable and Sustainable Energy Reviews, 113, 109226.

[26] B. Lin and T. Wiesner, 2019, An open source model of a parabolic trough solar field Computers and Chemical Engineering125 (2019) 514–531.

[27] D. Frejo, J. R., & F. Camacho, E. (2020). Centralized and distributed Model Predictive Control for the maximization of the thermal power of solar parabolic-trough plants. Solar Energy, 204, 190–199.

[28] López-Bautista, A. O., Flores-Tlacuahuac, A., & Gutiérrez-Limón, M. A. (2020). Robust model predictive control for a nanofluid based solar thermal power plant. Journal of Process Control, 94, 97–109. doi:10.1016/j.jprocont.2020.09.00

[29] Silva, R. N., Rato, L. M., & Lemos, J. M. (2003). Time scaling internal state predictive control of a solar plant. Control Engineering Practice, 11(12), 1459–1467.

[30] Karamali, M., & Khodabandeh, M. (2017). A distributed solar collector field temperature profile control and estimation using inlet oil temperature and radiation estimates based on Iterative Extended Kalman Filter. Renewable Energy, 101, 144–155.

[31] Bayas, A., Škrjanc, I., & Sáez, D. (2017). Design of fuzzy robust control strategies for a distributed solar collector field. Applied Soft Computing.

[32] Stephen Marsland Machine Learning: An Algorithmic Perspective, Second Edition (Chapman & Hall/CRC Machine Learning & Pattern Recognition) 2nd Edition October 8, 2014.

[33] The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics), Trevor Hastie, Robert Tibshirani, Jerome Friedman.

[34] Machine Learning (McGraw-Hill International Editions Computer Science Series) Paperback – October 1, 1997 by Tom M. Mitchell

[35] Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2020). A Survey of Optimization Methods From a Machine Learning Perspective. IEEE Transactions on Cybernetics, 50(8), 3668–3681. doi:10.1109/tcyb.2019.2950779

[36] Jitendra Malik, Learning to Optimize Ke Li, arXiv:1606.01885, 2016 and International Conference on Learning Representations (ICLR), 2017.

[37] Guillermo Martínez-Rodríguez, Amanda L. Fuentes-Silva, Daniel Velázquez-Torres, Martín Picón-Núñez, Comprehensive solar thermal integration for industrial processes, Energy, Volume 239, Part D,2022, 122332

[38] Razavi Zadegan, S. M., Mirzaie, M., Sadoughi, F. Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets. Knowledge-Based Systems 2013, 39, 133–143.

[39] Vibration control of uncertain multiple launch rocket system using radial basis function neural network

[40] Goodfellow, I., Bengio, Y., Courville, A., 2015. Deep Learning. The MIT Press, Cambridge, Massachusetts, USA.

[41] Fritsch F. N., Carlson R. E. Monotone Piecewise Cubic Interpolation. SIAM Journal on Numerical Analysis 1980; 17:238–246.

[42] Powell WB. From Reinforcement Learning to Optimal Control: A unified framework for sequential decisions. 2019, arxiv.

[43] Sutton, R., Barto, A., 2018. Reinforcement Learning: An Introduction, 2nd The MIT Press, Cambridge, Massachusetts, USA

[44] Nian, R., Liu, J., & Huang, B. (2020). A Review on Reinforcement Learning: Introduction and Applications in Industrial Process Control. Computers & Chemical Engineering, 106886.

[45] TRNSYS. Version 18. Thermal Energy System Specialists, LLC, USA.

[46] Anylogic simulation software, version 8.3.2, 2018, https://www.anylogic.com/

[47] MATLAB. Version R2021a. The MathWorks Inc., Natick, 2021.

[48] Solar resource maps of Ethiopia, 2017 The World Bank, Solar resource data: Solargis., https://solargis.com/maps-and-gis-data/download/ethiopia/, [accessed 14.09.18].

[49] Tilahun, F. B., Bhandari, R. and Mamo, M. Industrial process steam-consumption prediction through an Artificial Neural Networks (ANNS) approach, International Journal of Mechanical Engineering, 2: 72-81).

[50] Database of certified products for collectors, The Solar Keymark. http://www.solarkeymark.org/; 2017 [accessed 14.09.21].

[51] Tilahun, F. B., Bhandari, R., & Mamo, M. (2021). Design optimization of a hybrid solar-biomass plant to sustainably supply energy to industry: Methodology and case study. Energy, 220, 119736.

[52] Haeun Yoo, Ha Eun Byun, Dongho Han, Jay H. Lee, Reinforcement learning for batch process control: Review and perspectives, Annual Reviews in Control, 2021, https://doi.org/10.1016/j.arcontrol.2021.10.006.

[53] Navas, S. J., Rubio, F. R., Ollero, P., & Lemos, J. M. (2018). Optimal control applied to distributed solar collector fields with partial radiation. Solar Energy, 159, 811–819.

[54] Ernst D, Glavic M, Capitanescu F, Wehenkel L. Model predictive control and reinforcement learning as two complementary frameworks. Int J Tomogr Stat 2007; 6:122–7.

[55] Doerr, A., Nguyen-Tuong, D., Marco, A., Schaal, S., and Trimpe, S. Model-Based Policy Search for Automatic Tuning of Multivariate PID Controllers, 2017, In IEEE International Conference on Robotics and Automation.

[56] Bosch Center for Artificial Intelligence, Modeling a Physical System for System Control and Calibration, www.bosch-ai.com, accessed [19.11.2019]

[57] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. NIPS.

[58] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M., 2014. Deterministic policy gradient algorithms. ICML

[59] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv:1509. 02971.

[60] Schulman, J., Levine, S., Moritz, P., Jordan, M.I., Abbeel, P., 2015. Trust region policy optimization. ICML arXiv:1502.05477.

[61] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. Mach. Learn. arXiv:1707.06347..

[62] Lillicrap T, Hunt J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971

[63] Model-predictive control and reinforcement learning in multi-energy system case studies

[64] Turchi, C., 2010. Parabolic Trough Reference Plant For Cost Modeling With The Solar Advisor Model (SAM). National Renewable Energy Laboratory (NREL), Golden, CO.

[65] Stuetzle, T.A., 2002.Automatic Control of the 30MWeSEGSVI Parabolic Trough Plant. University of Wisconsin-Madison

[66] The African Monsoon Multidisciplinary Analyses (AMMA) program, https://database.amma-international.org/ [Accessed on 24.6.200].

[67] Zhang J., Zhao L. , Deng S., Xu W., Zhang Y.A critical review of the models used to estimate solar radiation, Renewable and Sustainable Energy Reviews 2017;70: 314–329.

[68] Hottel, H. C.A Simple Model for Estimating the Transmittance of Direct Solar Radiation Through Clear Atmospheres, Solar Energy, 1976; 18, 129.

[69] Yadav A.K. , Malik H. , Chandel S.S. Selection of most relevant input parameters using WEKA for artificial neural network based solar radiation prediction models, Renew. Sustain. Energy Rev. 2014; 31:509–519.

[70] Pazikadin, A. R., Rifai, D., Ali, K., Malik, M. Z., Abdalla, A. N., Faraj, M. A. Solar irradiance measurement instrumentation and radiation solar generation forecasting based on Artificial Neural Networks (ANN): A review of five years research trend. Science of the Total Environment 2020; 715:136848.

[71] K. Benmouiza, A. Cheknane. Forecasting hourly global solar radiation using hybrid k-means and nonlinear autoregressive neural network models. Energy Convers. Manag. 2013;75:561-569.

[72] Weipeng Cao, Xizhao Wang, Zhong Ming, Jinzhu Gao, A Review on Neural Networks with Random Weights, Neurocomputing, 2017; In Press, Corrected Proof — Note to users.

[73] Masoud Yaghinin, Mohammad M. Khoshraftar, Mehdi Fallahi. A hybrid algorithm for artificial neural network training. Engineering Applications of Artificial Intelligence, 2013:26:293-301.

[74] Diederik K., Ba J. Adam: A method for stochastic optimization. arXiv preprint 2014; arXiv:1412.6980.

[75] Kaplanis S, Kaplani E. Stochastic prediction of hourly global solar radiation for Patra, Greece. Appl Energy 2010; 87:3748–58.

[76] Ghimire, S., Deo, R. C., Raj, N., & Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. Applied Energy, 2019; 253, 113541.

[77] DeepMind AI Reduces Google Data Centre Cooling Bill by 40%, https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40, accessed [19.11.2019].

[78] Jakob N. F., Yannis M. A., Nando de F., Shimon W., Learning to Communicate with Deep Multi-Agent Reinforcement Learning, Cornell university, 2016, arXiv:1605.06676v2.

[79] A path to unsupervised learning through adversarial networks, 2016, FacebookEngineering, https://engineering.fb.com/ml-applications/a-path-to-unsupervised-learning-through-adversarial-networks/, accessed [19.11.2019].

[80] Lotter W., Kreiman G., and Cox D., Unsupervised Learning of Visual Structure Using Predictive Generative Networks, 2016, arXiv:1511.06380v2.

[81] Andrei A. R., Neil C. R., Guillaume D., Soyer H., Kirkpatrick J., Kavukcuoglu K., Pascanu R. and Hadsell R., Progressive Neural Networks,2016, Google DeepMind, London, UK.

[82] Ba L. J. and Caruana R., Do Deep Nets Really Need to be Deep?, Draft for NIPS 2014, arXiv:1312.6184v7.

[83] Forrest N. I., Han S., Matthew W. M., Ashraf K., William J. Dally2, Kurt Keutzer1, Squeezenet: Alexnet-Level Accuracy with 50x Fewer Parameters and <0.5mb Model Size, 2016, arXiv:1602.07360v4.

[84] Persson T., Fiedler F., Nordlander S., Bales C. and Paavilainen J, 2009, Validation of a dynamic model for wood pellet boilers and stoves, Applied Energy, 645-656.

[85] Thermal Energy Storage, Solar Heating and Cooling Systems, 2017, Elsevier Inc., http://dx.doi.org/10.1016/B978-0-12-811662-3.00004-9

[86] S. a. Klein and W. a. Beckman, "TRNSYS 16- Mathematical Reference," Trnsys 16, vol.5, p.484, 2007.

[87] Hochreiter, S. and Schmidhuber, J. (1995). Long short-term memory. Technical Report FKI-207-95, Fakult□at f□ur Informatik, Technische Universit□at M□unchen

[88] Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2019). Multivariate LSTM-FCNs for time series classification. Neural Networks, 116, 237–245.

[89] Berkenkamp, F., Turchetta, M., Schoellig, A.P., Krause, A., 2017. Safe model-based reinforcement learning with stability guarantees. NIPS 1–11.

[90] Jin, M., Lavaei, J., 2018. Stability-certified reinforcement learning: acontrol-theoretic prespective. ML arXiv:1810.11505.

[91] Bhatia, A., Varakantham, P., Kumar, A., 2018. Resource constrained deep reinforcement learning. ML arXiv:1812.00600.

[92] Ferreira, L., Riberiro, C., Bianchi, R., 2014. Heuristically accelerated reinforcement learning modularization for multi-agent multi-objective problems. Appl. Intell. 41, 551–562.

[93] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Sardinia (Italy): PMLR; 2010. p. 249–56.

[94] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision. Las Condes (Chile): IEEE; 2015. p. 1026–34.

[95] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.

[96] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data 2019;6(1):60.

[97] Nalepa J, Marcinkiewicz M, Kawulok M. Data augmentation for braintumor segmentation: A review. Front Comput Neurosci 2019;13:83.

[98] Branco P, Torgo Li, Ribeiro R. A survey of predictive modeling on imbalanced domains. ACM Comput Surv 2016;49(2):1–31.

[99] Yang Q, Zhang Y, Dai W, et al. Transfer learning. Cambridge: Cambridge University Press; 2020.

Author's Publications

1. F. B. Tilahun, M. Mamo and R. Bhandari, Design optimization of a hybrid solar-biomass plant to sustainably supply energy to industry: Methodology and case study. Energy, 2021, 119736. https://doi.org/10.1016/j.energy.2020.119736
2. F. B. Tilahun, M. Mamo and R. Bhandari, "Optimal Solar Field and Thermal Storage Sizing in Hybrid Solar Biomass Cogeneration Plant," *2020 IEEE PES/IAS PowerAfrica*, 2020, pp. 1-5, doi: 10.1109/PowerAfrica49420.2020.9219945, https://ieeexplore.ieee.org/document/9219945
3. F. B. Tilahun, M. Mamo and R. Bhandari, Design optimization and control approach for a solar-augmented industrial heating, Energy, Volume 179, 15 July 2019, Pages 186- 198, https://doi.org/10.1016/j.energy.2019.04.142
4. F. B. Tilahun, M. Mamo and R. Bhandari, Supply optimization based on society's cost of electricity and a calibrated demand model for future renewable energy transition in Niger. *Energ Sustain Soc* **9,** 31 (2019). https://doi.org/10.1186/s13705-019-0217-0
5. F. B. Tilahun, M. Mamo and R. Bhandari, Design Optimization and Demand Side Management of a Solar-Assisted Industrial Heating Using Agent-Based Modelling (ABM): Methodology and Case Study, E3S Web of Conferences 64, 02001 (2018), https://doi.org/10.1051/e3sconf/20186402001
6. F. B. Tilahun, M. Mamo and R. Bhandari, Industrial process steam-consumption prediction through an Artificial Neural Networks (ANNS) approach, International Journal of Mechanical Engineering, 2: 72-81), https://www.iaras.org/iaras/filedownloads/ijme/2017/012-0013(2017).pdf
7. F. B. Tilahun, M. Mamo and R. Bhandari, M. Economically realizable solar process heat solutions in Ethiopian textile industry with demand derived from artificial neural network data, WSEAS Transactions on Power Systems,2017;12:210-219, http://www.wseas.org/multimedia/journals/power/2017/a485916-070.pdf

Appendices

A. Machine Learning

A.1. A Brief History and Future Prospects

Machine learning (ML) algorithms are evolving from descriptive and diagnostic analytics to advanced application tools for prediction or prescription. However, this capability of machine learning is achieved through substantial cumulative efforts, of researches and developments. These efforts, generally, followed two different approaches, in two time periods, before and after the 1990s. A brief historical tour of machine learning, highlighting some useful and varied perspectives on the field is given next.

The notion of machine learning started in 1950 with the "Turing Test": a computer may have real intelligence, but only if it is be able to fool a human into believing it is also human. Since then, in the first period, initial works were focused towards a knowledge-driven ML using symbolic approaches it had inherited from artificial intelligence (AI). During the 1960's to 1980's, the role of ML was in design of neural network for computers (the perceptron) as well as formulations of explanation based learning (EBL) and pattern recognition, which were mostly made for achieving AI goals. In the 1990s, inclusion of techniques derived from statistics and probability theory shifted ML's focus to what is now known as a data-driven approach.

All of the earlier ML works that used the second approach can be classified as expert-based schemes. In these schemes, the characteristics and features in a dataset that enable streamlining the training process, supervised or unsupervised, were formulated or labeled by a human expert. Later on, the concept of deep learning, based on neural networks (NNs), was first coined by Geoffrey Hinton in 2006 —to convey the new capabilities of ML, for directly learning from unstructured datasets, in a scalable and autonomous way.

It is very difficult to anticipate the future roles of ML in detail. However, evident form the current trend in improved development frameworks as well as embedded target platforms and data access, there will be faster model training and evaluation of ML-based solutions. Thus, the interdisciplinary field of ML will continue to result in breakthroughs, as it has done since its inception, in different domains, including computer games, autonomous driving, predictive health care, and natural language processing. Accordingly, there are a variety of directed efforts

to discover new methods and learning paradigms in ML. Some examples of these trends, in the context of design and control optimization of complex physical systems, are:

Optimal control via agent-environment interaction: Recent efforts in these directions are being exploited, for example, for adaptive utilization of distributed resources (e,g 40% decrease in Google's datacenter cooling cost) [77] and creation of simulated environments (e.g. for generating large and low cost training data) [78]. Moreover, multi-agents with shared model or environment has also sparked research attention. This scheme allow representation of complex physical system in high-dimensional optimal control tasks.

Generative models: learn a probability distribution over training examples, and use unsupervised learning for predictive modelling. This framework is currently investigated for various applications such as simulating possible futures of a time-series [79], generalizing from small labeled datasets [80], and semi-supervised learning under few training data

Progressive learning: The aim is to train models that generalize for new task based on their knowledge of previous task [81]. Some recent research directions are for the ability to solve K independent tasks, accelerate learning through knowledge transfer and eliminate catastrophic forgetting. Progressive networks achieve this by training task specific model, while transferring previously learned knowledge through lateral connections.

Shallow models: Due to the scarcity, cost and speed of learning in deep neural models that require heavy training data, there is trend for realization of smaller models with comparable performance known as shallow models. Current research areas include training shallow networks by learning to mimic the performance of deep networks originally trained on large labeled training data [82], architectures with fewer parameters but equivalent performance to deep models [83].

A.2. Training Data and Algorithm Selection

Generally, discrete optimization methods for training ML algorithm can be used to assure tractability at a dimensionally low level, but they do not scale well as the complexity of the problem grow due to the “curse of dimensionality". Conversely, nonlinear function approximation models such as DNN can be used to remove the scalability issue, but they lack transparency and explainabilty due to their "black box" nature. This difficulty of understanding the intermediate representation could be a hindrance in realizing their full potential for safe and practical real-world solutions. Therefore, possible scalability and traceability implications of the ML solution on a given task are needed to be identified, usually at an individual level. This is the first challenge of algorithm selection, in which ML applications may be restricted to small scale and/or less sensitive tasks if explainability is required. In addition, with big datasets ML algorithm require large computational power, which may potentially lead to implementation constraints. In such cases simpler model alternatives that generalize and scale well should be explored.

There is also the issue of stability and safe operation when using ML for control application. Usually explicit models are used to find the stable region of attraction and initially guide the learning ML algorithm. On the other hand, augmenting the objective function by soft penalty factor may allow inclusion of state constraints for safe operation. Although these methods are efficient for physical systems whose underling processes are simple, there are needs for a rather advanced schemes, as we deal with nonlinear systems, in most cases, which can be time variant and uncertain. Even if it is possible to ensure stability of nonlinear systems, for example, by regulating the input-output gradient before algorithm update, the state constraint for safe operation is still a challenge. This is because the initially identified stable region of attraction increase in size as the learning agent explores the state-action space, and may eventually violate state constraints. Thus, designing an additional safety layer, which predict change in the constrained states, at each time step, may be required. Furthermore, the employed system identification technique for controller design should be correctly carried out to consistently represent process characteristics.

Furthermore, as it is not feasible to perform experiments on many industrial systems, a valid simulation environment that consistently represent process characteristics is often required for training ML algorithm for control application. Therefore, the operational dynamics of the system

are estimated through a system identification experiment. In this sense, the employed system identification technique should be correctly carried out to consistently represent process characteristics. Without an adequate model of the process to be controlled, even if it is realized by the controller, the performance flaw on the original system may be significant.

A vital part of algorithm development is evaluation of DNN model that aimed for a meaningful (desired accuracy) and robust (generalizing for real-world application) estimates [79]. To this end, the available data are often divided into training, validation, and test data, to train the model through an optimization process, estimate the generalization error of the trained model and tune model hyperparameters respectively. During this processes one of two undesirable situation might happen. The first happens when a model achieves a small training error but relatively large generalization error, and is known as overfitting. The second is the opposite of overfitting and is known as underfitting. Often, the possible cause of underfitting and overfitting is oversimplifying assumptions made when selecting DNN architecture or when the model becomes too sensitive to small variations or noise in the training data. Common approaches to address overfitting are using larger training datasets and controlling model complexity through regularization [80], dropout [81] and early stopping [82].

However, methods that alleviate these limitations are in development. Commonly used techniques are data augmentation and transfer learning.

Data augmentation: While usage of large datasets help to avoid overfitting, it might not be practically feasible when the available valid data is small or is not accessible in a timely and cost effective way. In these situations, computational method known as data augmentation, which uses noise to generates new samples from existing ones, can be used in deep learning [83].

Transfer learning: Similar to data augmentation method, transfer learning has also been widely used when obtaining large datasets is often impractical. Transfer learning refers to the process of using rule or knowledge acquired from another domain to solve related but not the same problems in a target domain under limited data. For instance, in forecasting, such as wind speed, transfer learning may be used for sharing common features collected from different farms to improve prediction accuracy [84]. Transfer learning methods can be instance-based that reuses data samples, feature/relation-based that transform the data features/relations of source and target domain into a unified feature space or model-based that uses pre-trained models to extract latent

features [85]. Owning to its effectiveness, model-based transfer learning is widely used for practical applications.

Approach in This Work

Machine learning algorithms are independent of the area of knowledge that they refer to. However, in general, algorithms with stochastic parameters are required to solve complex tasks in, and for systems which consist of distributed assets that interact nonlinearly with each other, in a changing environment, and are therefore, heterogeneous in nature and unpredictable in essence. These kinds of systems as well as the ML algorithms can be regarded as complex systems, and need to be dealt with as such.

The inclusion of efficient data tactics (e.g. metadata and data fusion) has created the possibility of performance lift, which can invert the existing approach to selecting powerful algorithm that require bigger and better computing. In general, such computing capability may not be available to industries in practice. Furthermore, the gains from such powerful algorithms comes with implementation cost and/or scalability trade-offs, which makes the usefulness of this approach even more doubtful. Additional requirement such as using simpler architecture and configuration, to reduce complexity (speed of computation and analysis) and simplify hardware embedding [8], are also equally important. In this sense, there is a need to check accuracy and tolerances of various algorithms and model architectures for the handling the same task. Other model complexity reduction techniques such as handling inputs differently through feeding filtered derivatives of inputs to, for example, a CNN for a simpler RNN replacement should also be considered. Even after determining the model architecture, there is still a need to optimally configure it (e.g. the number of hidden layers and neurons in those layers of a NN). This optimally configuring task is often challenging as it requires complex procedures like, for example, "pruning" to eliminate hidden layers and/or neurons in NNs.

In order to address the aforementioned issues, this research work formulate the optimal algorithm selection in ML as a data-driven decision process (DDP) through automated design of experiment. In this scheme the test and validation also use a data-driven approach, in order to remove biases from the variety of available measures and their data size dependent performances. The steps of the design experiment are:

Active dataset selection: Due to the high computation and replication experiments there is a need to work with a smaller data samples to get faster results. At the sometime, it is also necessary to select samples that lead to reliable results i.e. are active datasets.

Define the ML interfaces and their performance measures: In this research work different ML techniques that range from clustering to predictive modelling and reinforcement learning, each with different performance measures that suits their application are considered. To this end, it is necessary to identify both the various ML algorithms and models doing the same task as well as possible ways of measuring their performances.

Simulation and replication experiments: Reusable scripts that automate the design experiment are desirable to facilitate the selection procedure.

B. Generic Models

The distributed energy resources (DERs) that are modeled as generic are the solar field, heat exchanger, biomass boiler, storage and industrial steam turbine. The practical requirement of these DERs is to convert, store or convey thermal energy. For these applications we have different DER technologies and different types of same technology. However, all these DERs can be modelled similarly, because each DER has similar, but not identical, characteristics. The model can then be parameterized differently to represent heterogeneous DER models in industrial solar thermal system. These models are described as follows.

B.1. Distributed Solar Field (DSF)

The useful amount of heat collected from the DSF, $\dot{Q}_{net}$ can be determined by from:

$$\dot{Q}_{net} = \dot{Q}_{abs} - \dot{Q}_{msl} \quad (B.1)$$

$$\dot{Q}_{abs} = A_{eff} DNI \eta_{sf} \quad (B.2)$$

Where $\dot{Q}_{abs}$, η_{sf} I_{DNI} and A_{eff} represent the absorbed solar power, absorber efficiency, solar irradiation and effective total area of collectors in DSF. $\dot{Q}_{msl}$ comprises thermal losses (piping and expansion vessel). The solar field outlet temperature can be solved iteratively (after assigning initial values based on the design assumption) as:

$$T_{sf,out} = T_{sf,in} + \dot{Q}_{net} / \dot{m}_{htf,sf} \rho_{htf} c_{htf} \quad (B.3)$$

The absorber efficiency can be calculated using

$$\eta_{sf} = \eta_o - a_1 \frac{(T_m - T_a)}{G} - a_2 \frac{(T_m - T_a)^2}{G} \quad (B.4)$$

Where $\eta_c, G, T_m, T_a, \eta_o, a_1, a_2$ denote maximum efficiency (at no heat loss), direct (or global) irradiance, mean HTF temperature in DSF, 1st and 2nd order heat loss coefficient respectively. A parameter identification technique as described in [65] can be used for evaluating these parameters.

Heat Exchanger (HEX)

The source side HEX outlet temperature dynamics depend on the source side inlet as well as load side inlet and outlet temperature, which can be given as:

$$\rho C_F V_{HE} \frac{dT_{HE,s,out}(t)}{dt} = a_{HE}\rho C_F \dot{V}_F \left(T_{HE,l,out}(t) - T_{HE,s,out}(t)\right) - L_{HE} Q_{g,HE} \quad (B.5)$$

Where a_{HE} is the coefficient and the heat gained by the HEX, $Q_{g,HE}$is given as:

$$Q_{g,HE} = h_{HE} A_{HE}\left(T_{HE,s} - T_{HE,l}\right) \quad (B.6)$$

Where $T_{HE,s}\ and\ T_{HE,l}$ are the average source side and load side temperatures in the HEX. The heat transfer coefficient, h_{HE}, depends on the exchanged fluids flow rate as [65]:

$$h_{HE} = \delta_{HE}\left({\dot{V}_F}/{\dot{V}_{F,r}} + \frac{\dot{m}_W}{\dot{m}_{W,r}}\right) \quad (B.7)$$

Where $\dot{m}_W, \dot{m}_{W,r}$ are the instantaneous and reference mass flow rate of the working fluid and δ_{HE} is the HEX coefficient that depends on density and heat capacity of the fluids. The temperature of the steam is calculated from the heat exchanger effectiveness as follows [65]:

$$\in (t) = \frac{T_{HE,l,out}(t) - T_{HE,l,in}(t)}{T_{HE,s,in}(t) - T_{HE,l,in}(t)} = \alpha_1 + \alpha_2 h_{HE} \quad (B.8)$$

On the other hand, the load side heat exchanger outlet temperature dynamics is given by [65]:

$$\frac{dT_{HE,l,out}(t)}{dt} = \frac{1}{\tau_\in}\left[T_{HE,l,in}(t) - T_{HE,l,out}(t) + \in \left(T_{HE,s,in}(t) - T_{HE,l,in}(t)\right)\right] \quad (B.9)$$

Where $\tau_\in$ is the time constant of the entire heat exchanger system.

Biomass Boiler (BB)

This model is necessary, since available modelling tools such as TRNSYS does not support HTF other than water/steam. In such cases, the biomass boiler can be model in two stages: combustion and heat exchanger. In the combustion stage fuel and air is combusted adiabatically to form a hot gas with mass flow rate and a temperature as:

$$\dot{m}_g = \dot{m}_f + \dot{m}_a \quad (B.10)$$

$$T_{ad} = T_{amb} + \frac{\dot{Q}_{cmb} + P_e}{\dot{m}_g C_g} \quad (B.11)$$

The air factor λ that control the required air for combustion can be given as [84]:

$$\lambda = 2.0395\gamma^{-0.4415} \quad (B.12)$$

Where the constant γ is calculated by:

$$\gamma = \frac{\dot{Q}_{cmb}}{\dot{Q}_{cmb}^{max}} \tag{B.13}$$

The required air flow is then determined as:

$$\dot{m}_a = \frac{\lambda r_{a,f} \dot{Q}_{cmb}}{LHV_b} \tag{B.14}$$

Where $r_{a,f}$, LHV_b are the air-to-fuel ratio and lower heating value of the biomass. The efficiency of transferring flue gas temperature to the HTF is calculated as [84]:

$$C_g = (1.050304 + 0.000067T_{f,g} - 0.180080\omega_{O_2} - 0.000218T_{f,g}\omega_{O_2} \tag{B.15}$$

Whereω_{O_2} is O_2 volume fraction in dry gas.

The last part of the model describes the heat exchanger where the thermal energy of the flue gas is given to the HTF. The inlet and outlet HTF temperature T_{HTF} can be determined from the heat exchanger effectiveness as follows [65]:

$$\varepsilon(t) = \frac{T_{HTF}^{o}(t) - T_{HTF}^{i}(t)}{T_{f,g}(t) - T_{HTF}^{i}(t)} \tag{B.16}$$

Thermal Energy Storage (TES)

The TES model is described by the following equation,

$$\frac{dQ(t)}{dt} = \dot{Q}_{in}(t) - \dot{Q}_{out}(t) - \dot{Q}_{loss} \tag{B.17}$$

where $Q, \dot{Q}_{in}$ and$\dot{Q}_{out}$ denote energy, effective in-and out-flowing thermal power. $\dot{Q}_{\text{loss}}$ is the losses over time when TES is in the static mode (no in- or out-flows). The following valid assumptions are considered while modelling these losses [85]:

- Sufficient TES insulation is assumed, therefore heat transfer rate to ambient is negligible.
- Steady and slight increase in ambient temperature is also assumed, leading to a constant average temperature difference in TES within each time step t_i (from i to i + 1).

Therefore, $\dot{Q}_{\text{loss}}$ can be modelled as an average value over the whole period t as:

$$\dot{Q}_{\text{loss}} = \frac{V_{\text{st}}\rho_{\text{st}}c_{\text{st}}\left(\text{T}_{\text{st\_av, end}} - \text{T}_{\text{st\_av, sart}}\right)t}{\sum_{i=\text{Start}}^{\text{End}} t_i\left(\text{T}_{\text{amb,i}} - \text{T}_{\text{st\_av, i}}\right)} \quad (B.18)$$

Conversion losses in TES can be added by an efficiency $\eta_{\text{st, in}}$

$$\dot{Q}_{in} = \eta_{\text{st, in}}\,\dot{Q}'_{\text{in}} \quad (B.19)$$

which describes the relationship between the real in-flow thermal power $\dot{Q}'_{\text{in}}$ and the effective inflow. Analogously, the actual output thermal power $\dot{Q}'_{\text{out}}$ can be defined as

$$\dot{Q}'_{\text{out}} = \eta_{\text{st, out}}\,\dot{Q}_{out} \quad (B.20)$$

It is also possible to use the specific heat transfer coefficient values of TES from the manufacturer.

Conventional Thermal Plant

When the STS is designed to work as a cogeneration plant, the turbine can be modelled using Stodola's law of the ellipse, to determine the inlet pressure based on the outlet pressure, the steam mass flow rate and reference values of inlet and outlet pressure and mass flow rate. The enthalpy and pressure values are computed using an isentropic efficiency, η_{is}, as [86]:

$$\eta_{is} = \eta_{is,ref} \cdot (1 + a \cdot f_{is} + b \cdot f_{is}^2 + c \cdot f_{is}^3) \quad (B.21)$$

Where

$$f_{is} = \frac{\dot{m} - \dot{m}_{\text{ref}}}{\dot{m}_{ref}} \;;\; -0.7 \leq f_{is} \leq 0.7 \quad \text{(B.22a)}$$

$$h_{\text{out}} = h_{\text{in}} - \eta_{\text{is}} \cdot \left(h_{\text{in}} - h_{in,is}\right) \quad (B.22b)$$

Thus, the power generated by the turbine is

$$P_{\text{gen}} = m_{\text{in}} \cdot \eta_{\text{gen}} \cdot \left(h_{\text{in}} - h_{\text{out}}\right) \quad (B.23)$$

If there are reheating processes of the steam after the first expansion, then it is necessary to repeatedly utilize the above turbine stage model.

C. Deep Neural Network Models

A Multi-layer perceptron (MLP), having multiple layers, each with one or more artificial neurons is shown in Figure 2. For MLP information from the input propagate from neuron to neuron and through layer by layer up until the output layer that process and gives the final value. Prior to their usage as a useful network, MLPs are trained and tested on valid datasets. The training process iteratively adjust connection weights and biases, which are often initialized randomly, and eventually find their optimal values based on a loss function. The ability of MLPs to learn complex nonlinear functions depends on the type of activation functions. Common nonlinear activation functions are Rectified linear unit (ReLU) and its variants, Sigmoid, and $Tanh$. In general ReLUs are used in hidden layers whereas Sigmoid and $Tanh$ function are used in the output layers of networks designed for classification or regression.

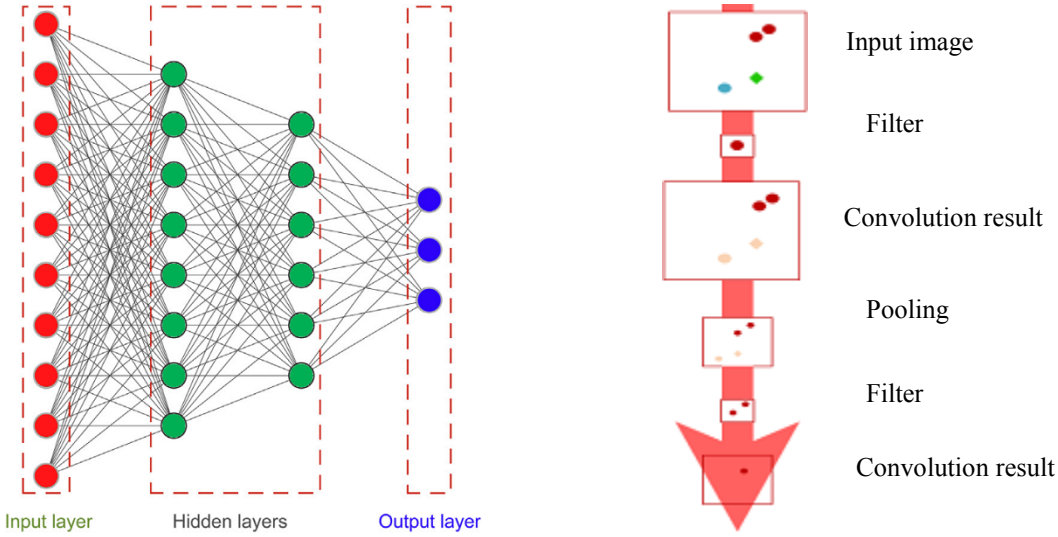


Figure C.1. Fully connected networks variant as MLP (left) and CNN (right).

MLPs result performance improvement as the number of layers and complexity of interconnection increases. This is often considered to be the reason for designing deep network architectures to handle complex tasks.

Deep learning architectures

The deep neural network (DNN) architectures may be divided in to two subgroups, namely, as feed forward and recurrent neural networks (RNN). The first category consists of fully connected networks (FCN) which are interchangeably known as MLPs and convolutional neural networks.

On the other hand typical example of RNN is the long short-term memory (LSTM). These DNNs are explained as follows

Convolutional Neural Network (CNN): Using MLP to solving complex task with large number of inputs requires huge training datasets, which in practice might not be available. CNNs, which are basically MLPs with neurons replaced by convolutions and pooling operators, can handle such task through sparse connectivity and parameter sharing. Pooling layers do not have coefficients or weights, but downsample the data by a factor, enabling CNNs add more information. As a result it is it possible to achieve a better performance and lesser memory requirements compared with MLPs. For example consider the task of identifying locations of the diagonally oriented red dots (FigureC.1). A CNN with two convolution layers and two filters can solve this problem. The task of the first layer is to identify red dots by applying the first filter everywhere in the image, and resulting in high responses (dark red) where the image is most similar to the filter. On the other hand, the second layer detect the high response of the first layer, in a diagonal orientation using anther filter. By stacking several convolutional-pooling layers this way, CNNs can be used to handle complex patterns identification and regression tasks.

Recurrent Neural Network (RNN): In general, RNNs are used for tasks with sequential data that require processing current as well as prior inputs across time. Thus an RNN, as depicted in Figure3.5a, inputs the first instance of a sequence to make a prediction, and then adds its own output with the next instance of the sequence for successive predictions.

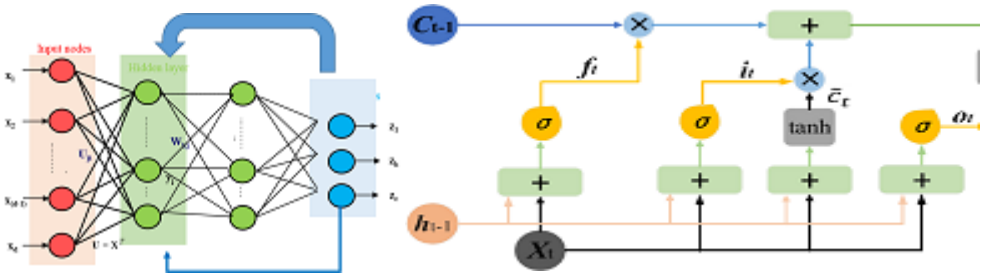


Figure C.2. Schematic of a 2-layer RNN (left) and LSTM network architecture (right).

However, sequential prediction from RNN is often characterized by historical information loss. To address this problem, an RNN variant known as long-short term memory (LSTM) network is proposed. An LSTM network, uses a variety of well-designed gates, to continually learn new tasks as well as memorize how to implement it in the future (FigureC.2). The LSTM architecture was originally introduced by Hochreiter and Schmidhuber [87] with the purpose of overcoming

the vanishing or exploding gradients problem [88], by allowing constant error flow through self-connected units embedded in the LMST cell. This key feature of LSTM makes it capable of learning long-term dependencies, as opposed to other RNN variant.

Memory mechanism of LSTM starts with a forget gate that determines what information to discard from the previous memory cell and is given by

$$f_t = \sigma\left(w_{fh}^T \times h_{t-1} + w_{fx}^T \times x_t + b_f\right) \quad (\mathrm{C}.1)$$

On the other hand, input gate determines what information to keep in current memory cell, which can be expressed as

$$i_t = \sigma(w_{ih}^T \times h_{t-1} + w_{ix}^T \times x_t + b_i) \quad (\mathrm{C}.2)$$

A state update mechanism of memory cell combines previous and current information as

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (\mathrm{C}.3\mathrm{a})$$

where

$$\tilde{c}_t = \tanh(w_{ch}^T \times h_{t-1} + w_{cx}^T \times x_t + b_c) \quad (C.3b)$$

Output gate gives out information from the memory block based on the relation given by

$$o_t = \sigma(w_{oh}^T \times h_{t-1} + w_{ox}^T \times x_t + b_o) \quad (C.4)$$

Finally current memory block of LSTM unit is updated as

$$h_t = o_t \times \tanh(c_t) \quad (C.5)$$

In the above LSTM memory mechanism, f_t, i_t, c_t, h_t, o_t denote the outputs of forget gate, input gate, temporary memory cell, new memory cell, output gate, and memory block at time point t, respectively. $w_{fh}, w_{ix}, w_{ch}, w_{fh}, w_{ho}, w_{ax}$ denote the corresponding weights, by which information of the previous memory cell h_{t-1} and the input of current state x_t can be properly traded off. $b_{f\ldots o}, \sigma$ are biases and sigmoid activation function.

D. Case Study Data

Table D.1. Performance parameters of APSE-30 ETC.

Performance related to aperture	Par.	η0	a_1	a_2	$k_{\theta d}$
	Value	0.710	1.737	0.008	1.382
Incident angle modifier	Angle	20°	30°	40°	60°
	$k_{\theta b}$	1.00	0.99	0.98	0.94
	Angle	20°	30°	40°	60°
	$k_{\theta b}$	1.02	1.08	1.33	1.51
Collector area (Aperture)	2.84m^2				
Effective thermal capacity	94.3KJ/(m^{2}K)				
Flow rate (per aperture area)	0.02 Kg/(Sm^2)				

Table D.2. Specification of electrical boiler

Particular	**Unit**	**Electrical boiler**
Power consumption	kWh/day	13200
Steam pressure	bar	5
Steam temperature	^{0}C	152
Steam enthalpy (base on steam table)	kCal/kg	657.4
Feed water temperature	^{0}C	65
Consider boiler efficiency	%	95
Steam generation	kg/day	18204.6
Specific energy consumption	kWh/kg	0.725

Table D.3. PTC configuration parameters for the reference DSF

Solar field	**Configuration parameters**
Location	Daggett, CA
Angle of incidence	11.4136°
Heat transfer fluid	Synthetic oil (Therminol VP-1)
Number of solar collectors	1208
Number of loops	302
Number of SCA per loop	4
Number of mirror modules per SCA	12
Aperture width & SCA aperture area	6m, 522m^2
HTF inlet & outlet temperature	293^{O}C, 393^{O}C
Flow rate per loop	1-7kg/s